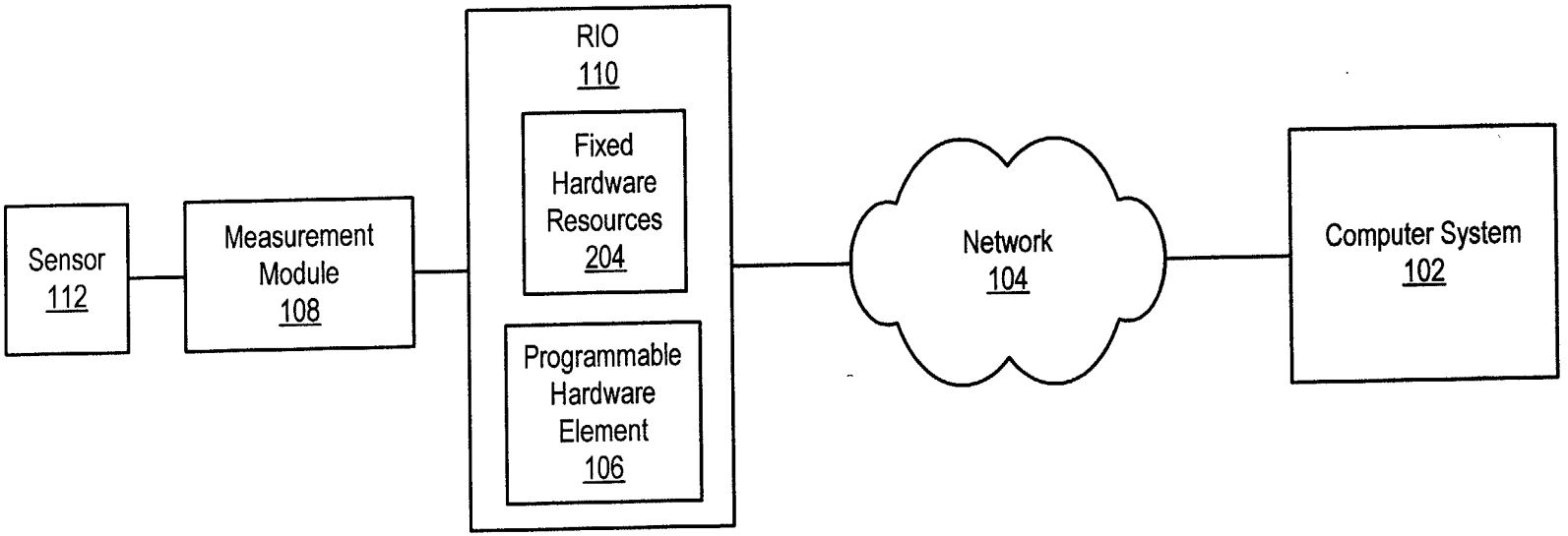

RIO
110
Fixed
Hardware
Resources
204
Programmable
Hardware
Element
106
Sensor
112
Measurement
Module
108
Network
104
Computer System
102

Figure 1

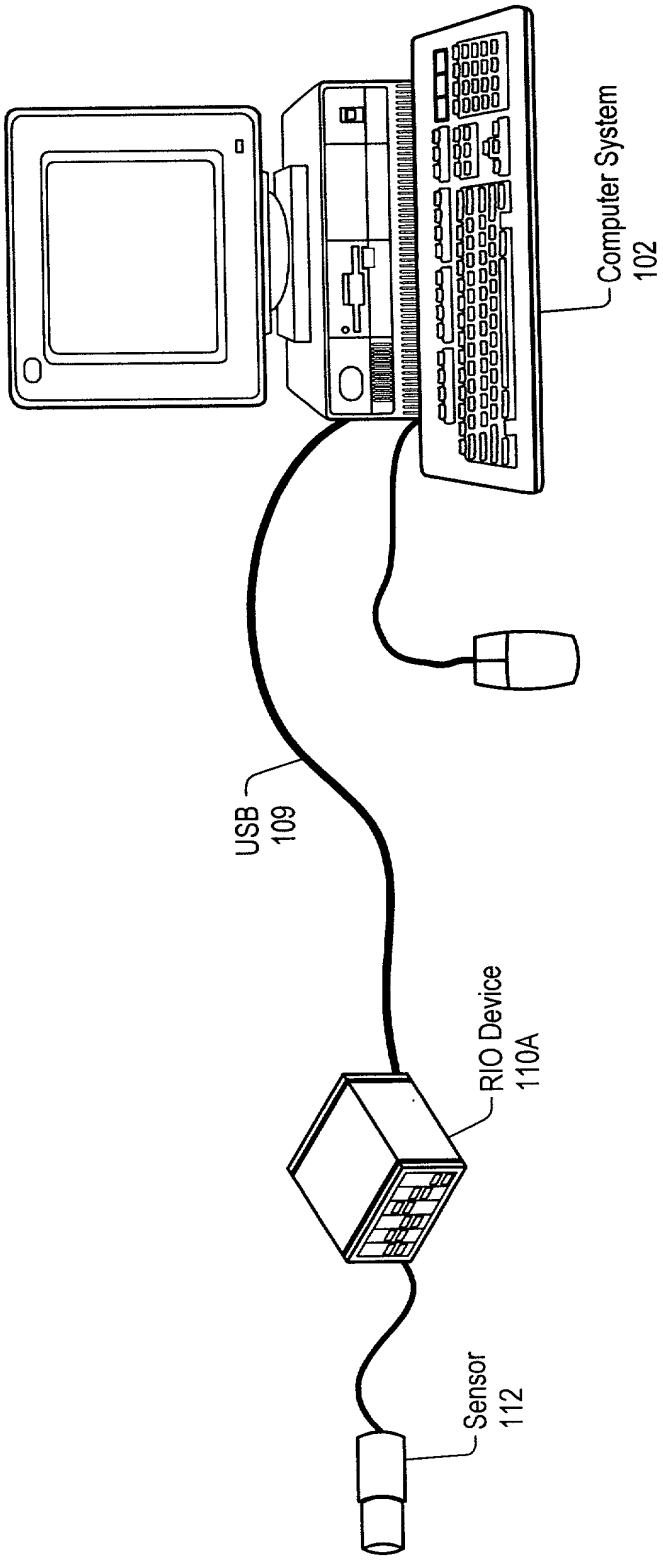


Figure 1A

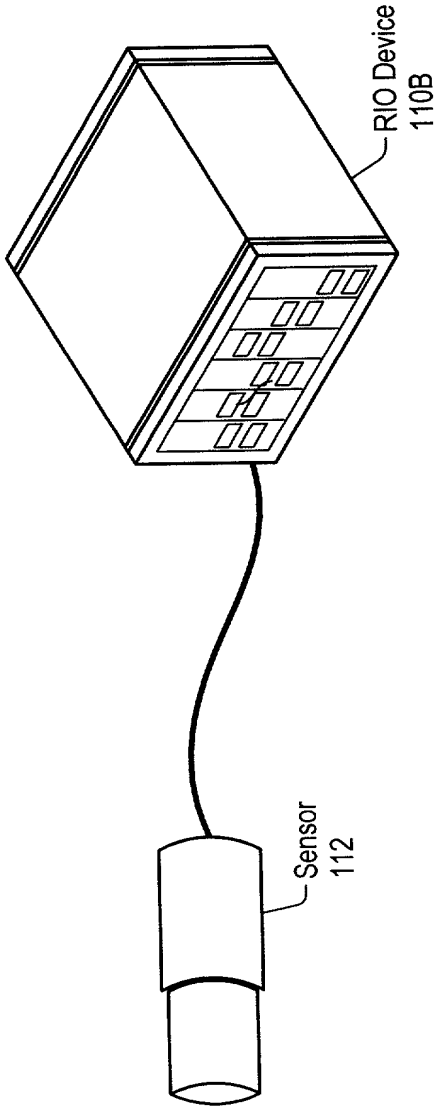


Figure 1B

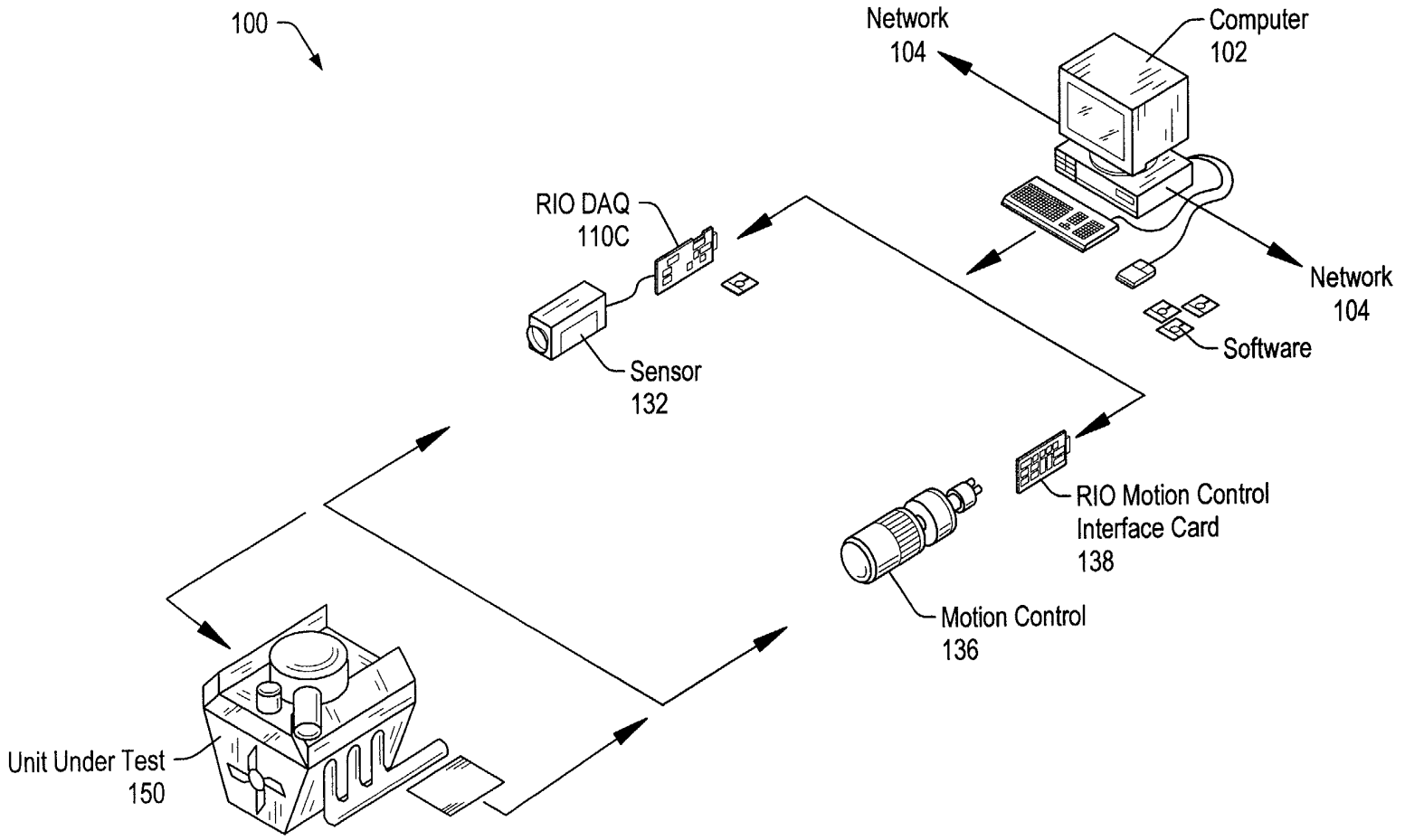

100
Network
104
Computer
102
RIO DAQ
110C
Network
104
Software
Sensor
132
RIO Motion Control
Interface Card
138
Motion Control
136
Unit Under Test
150

Figure 1C

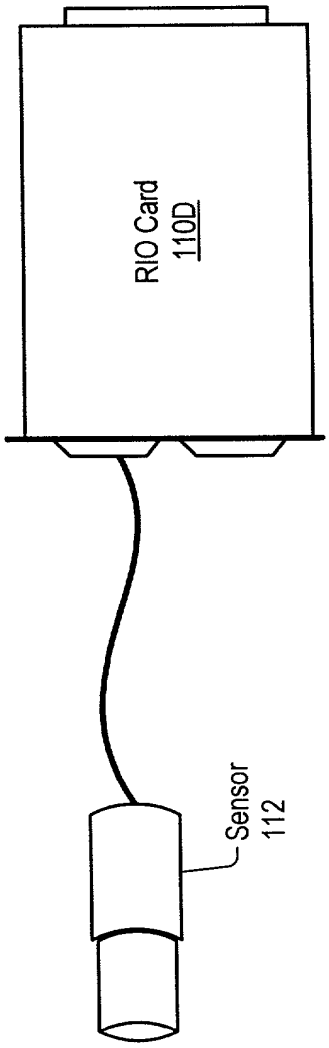


Figure 1D

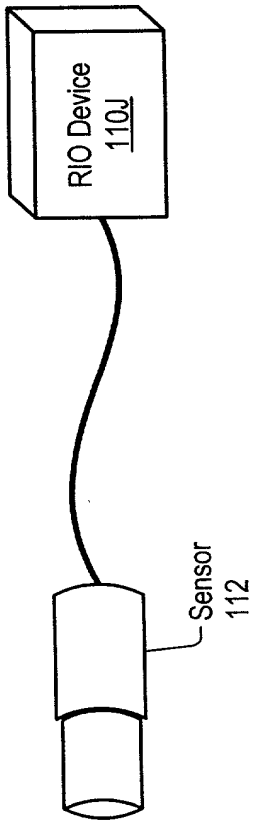


Figure 1E

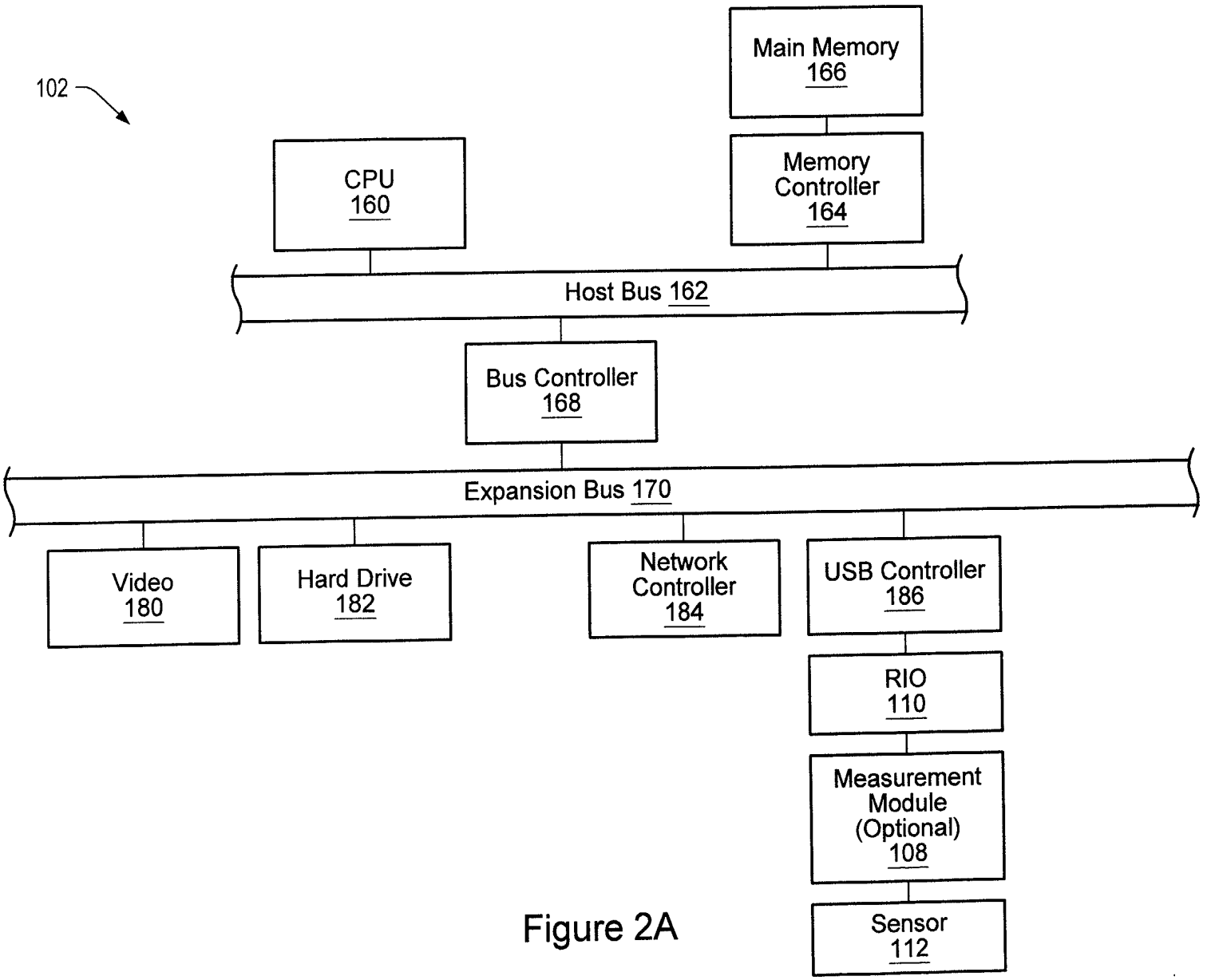

102
Main Memory
166
CPU
160
Memory
Controller
164
Host Bus 162
Bus Controller
168
Expansion Bus 170
Video
180
Hard Drive
182
Network
Controller
184
USB Controller
186
RIO
110
Measurement
Module
(Optional)
108
Sensor
112

Figure 2A

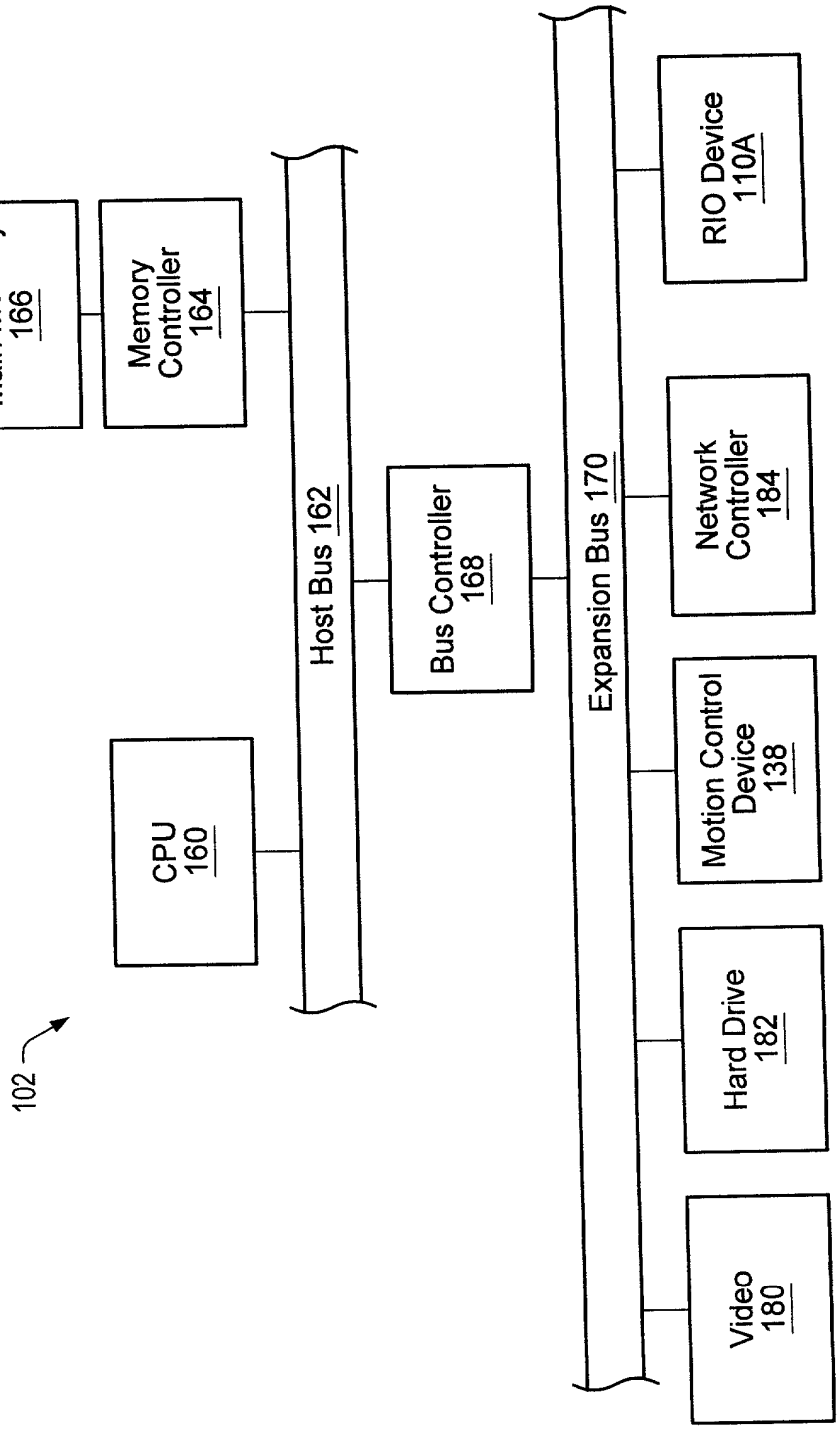

102
CPU 160
Main Memory 166
Memory Controller 164
Host Bus 162
Bus Controller 168
Expansion Bus 170
Video 180
Hard Drive 182
Motion Control Device 138
Network Controller 184
RIO Device 110A

Figure 2B

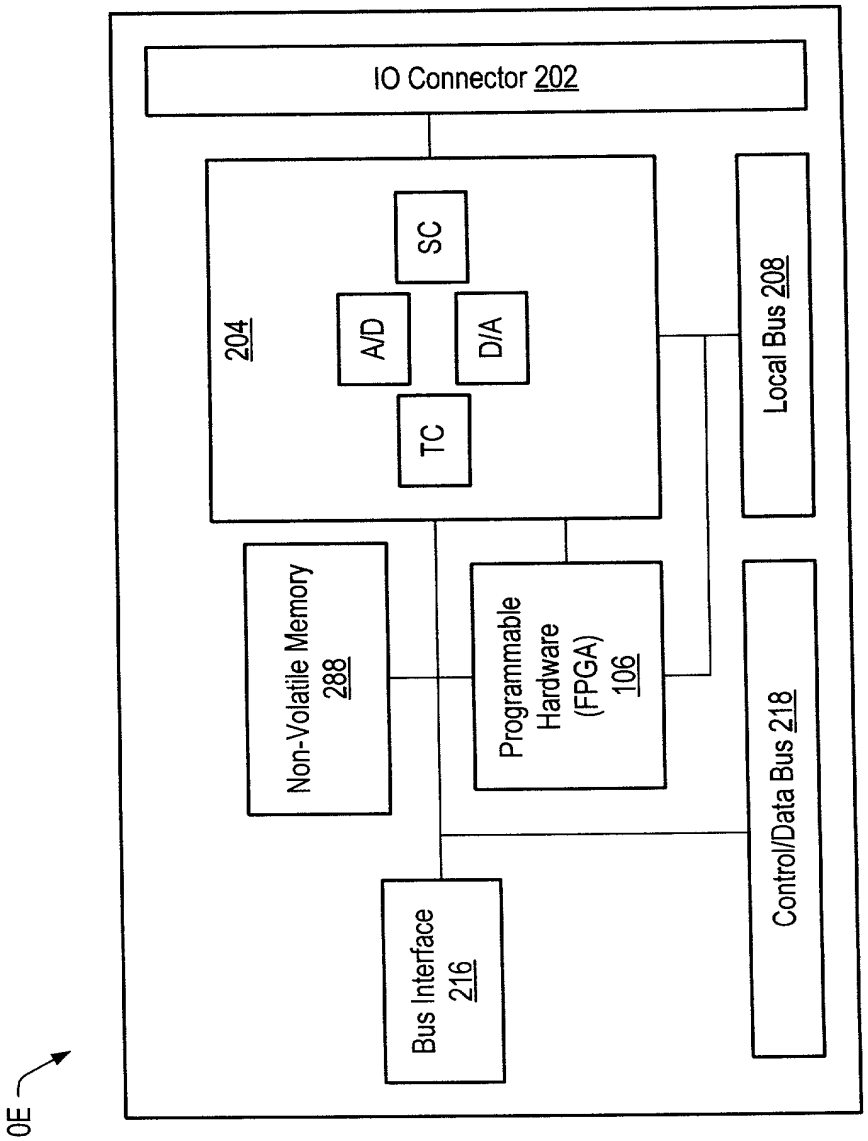

110E
IO Connector 202
204
SC
A/D
D/A
TC
Local Bus 208
Non-Volatile Memory 288
Programmable Hardware (FPGA) 106
Control/Data Bus 218
Bus Interface 216

Figure 3A

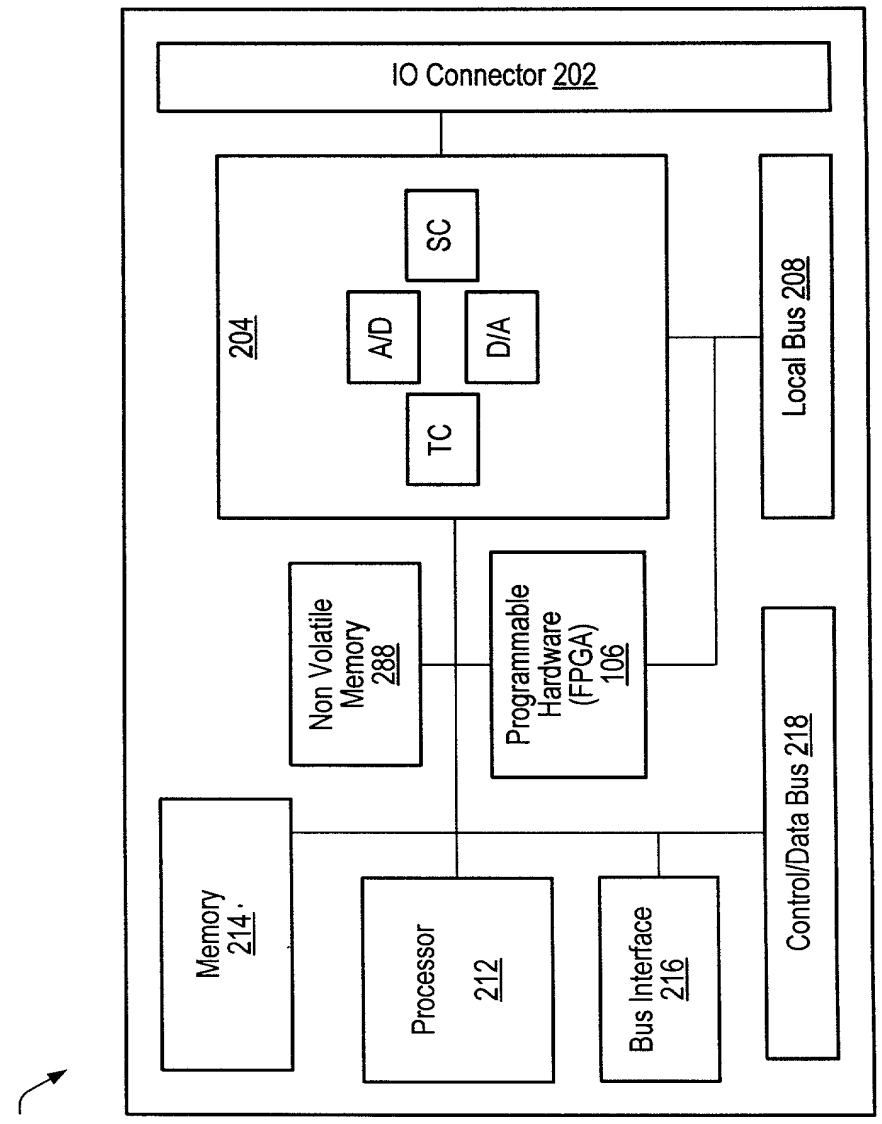

110F
IO Connector 202
204
SC
A/D
D/A
TC
Local Bus 208
Non Volatile Memory 288
Programmable Hardware (FPGA) 106
Memory 214
Processor 212
Bus Interface 216
Control/Data Bus 218

Figure 3B

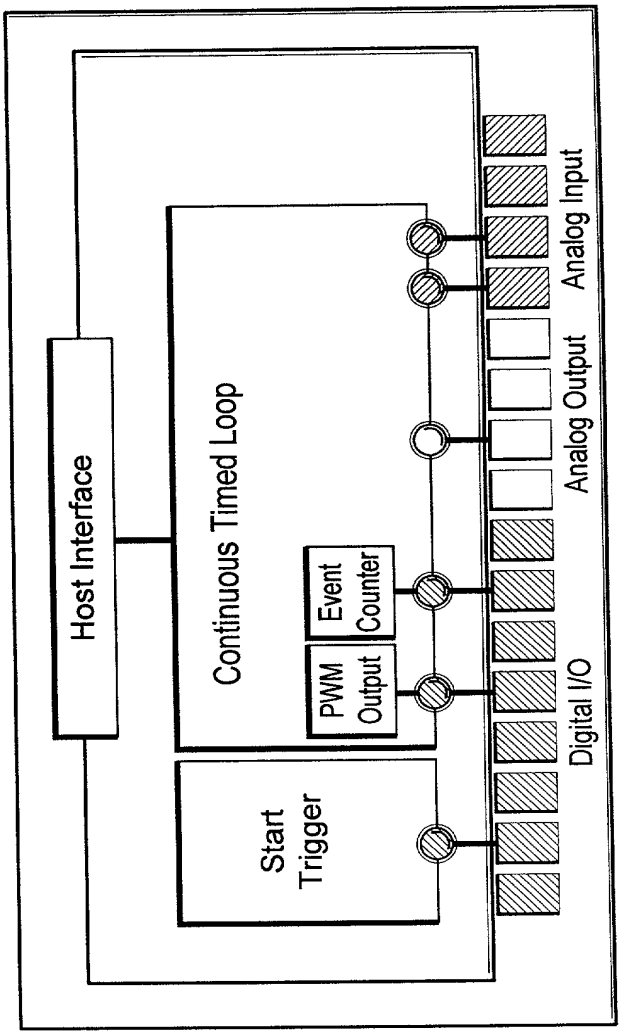

Host Interface
Continuous Timed Loop
Start Trigger
PWM Output
Event Counter
Digital I/O
Analog Output
Analog Input

Figure 3C

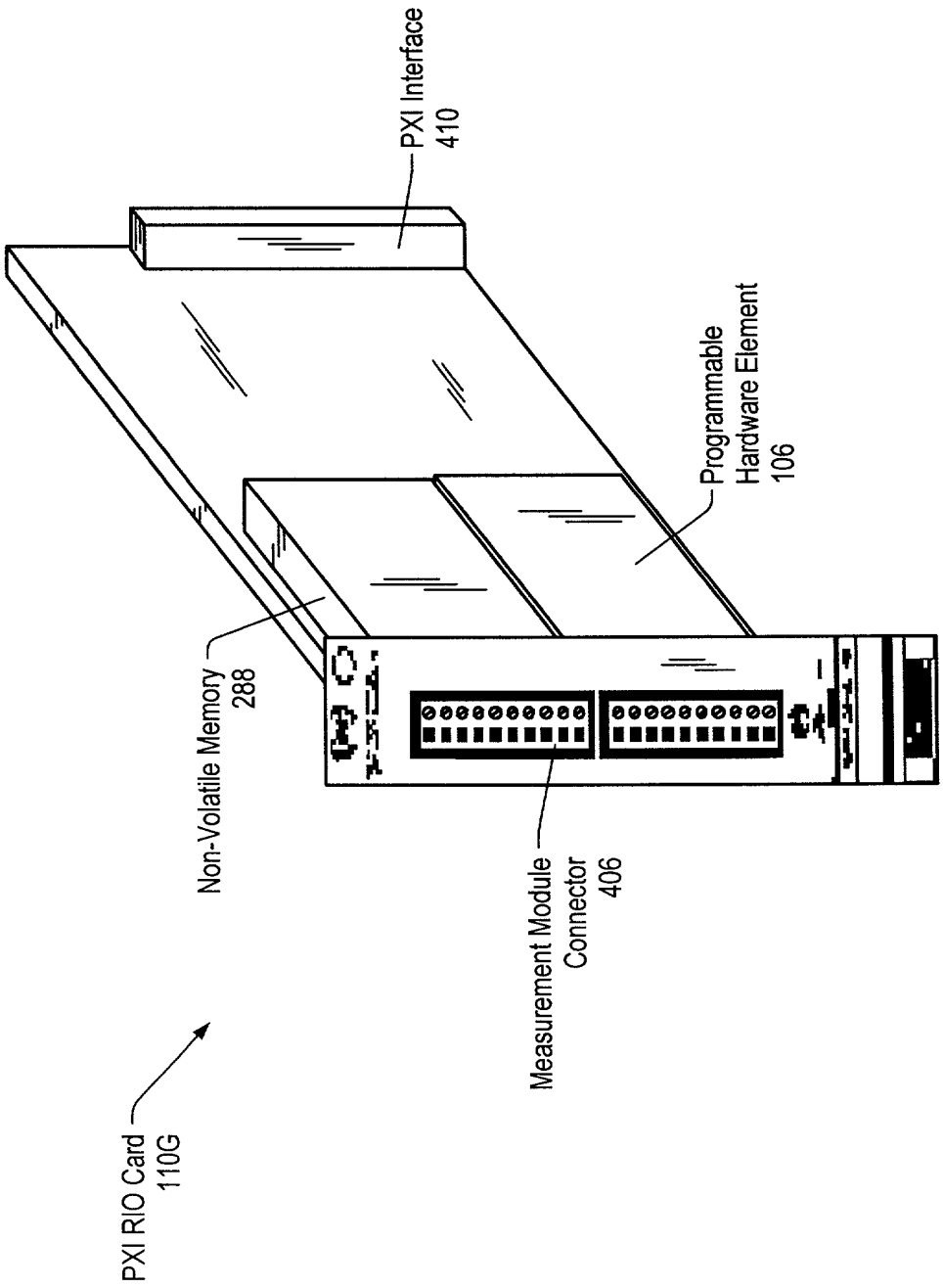

PXI Interface
410
Programmable
Hardware Element
106
Non-Volatile Memory
288
Measurement Module
Connector
406
PXI RIO Card
110G

Figure 4

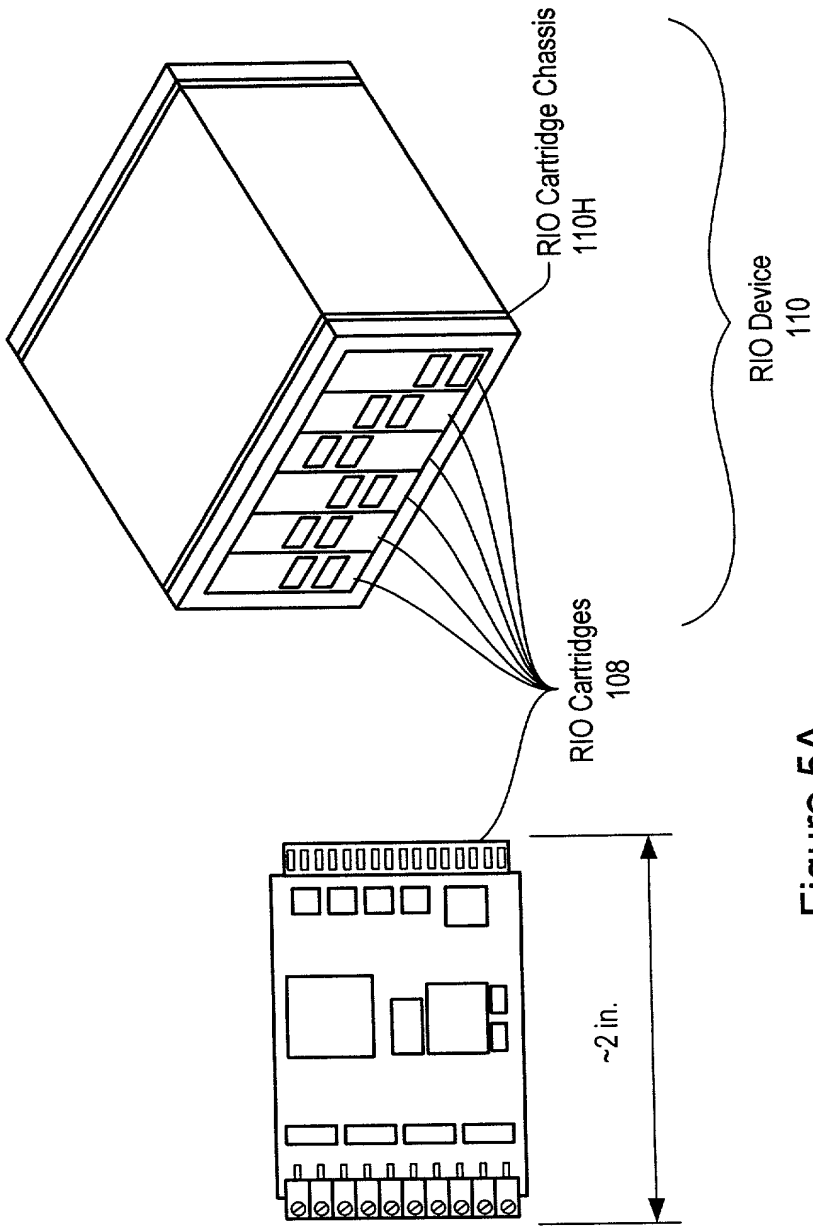

RIO Cartridge Chassis
110H
RIO Device
110
RIO Cartridges
108
~2 in.

Figure 5A

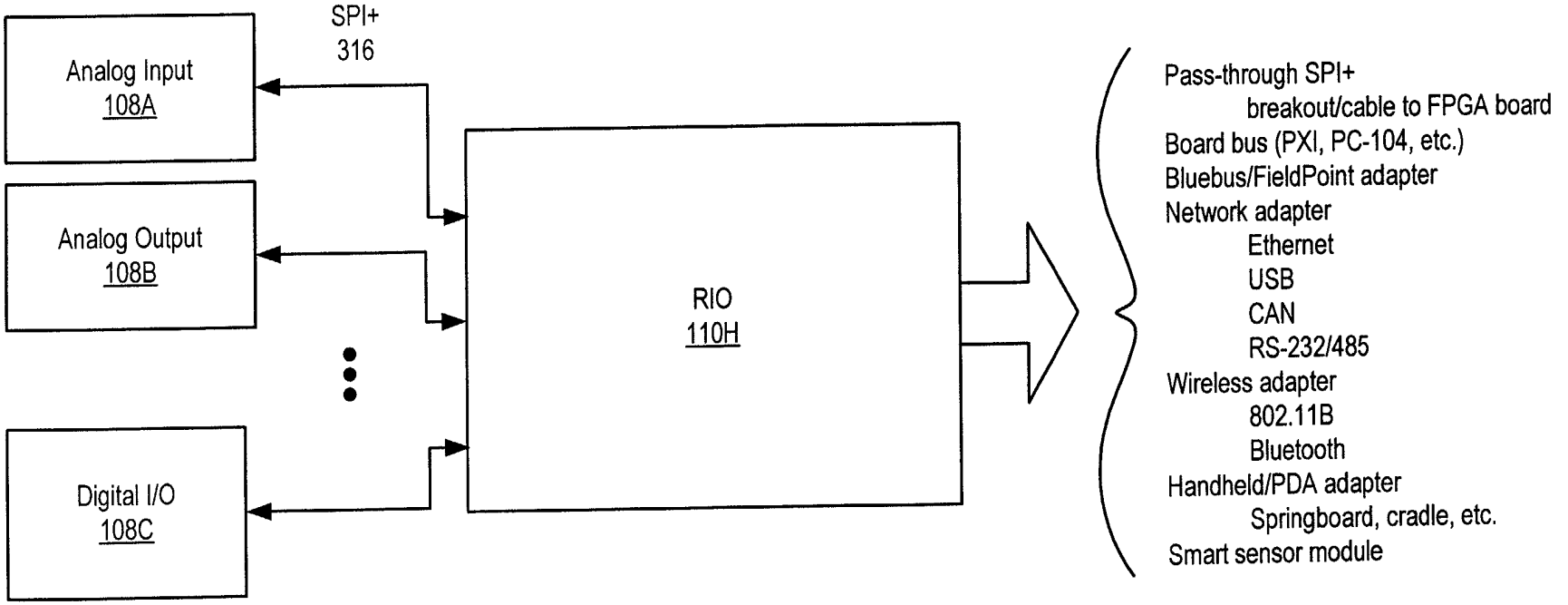

SPI+
316
Analog Input
108A
Analog Output
108B
Digital I/O
108C
RIO
110H
Pass-through SPI+
breakout/cable to FPGA board
Board bus (PXI, PC-104, etc.)
Bluebus/FieldPoint adapter
Network adapter
Ethernet
USB
CAN
RS-232/485
Wireless adapter
802.11B
Bluetooth
Handheld/PDA adapter
Springboard, cradle, etc.
Smart sensor module

Figure 5B

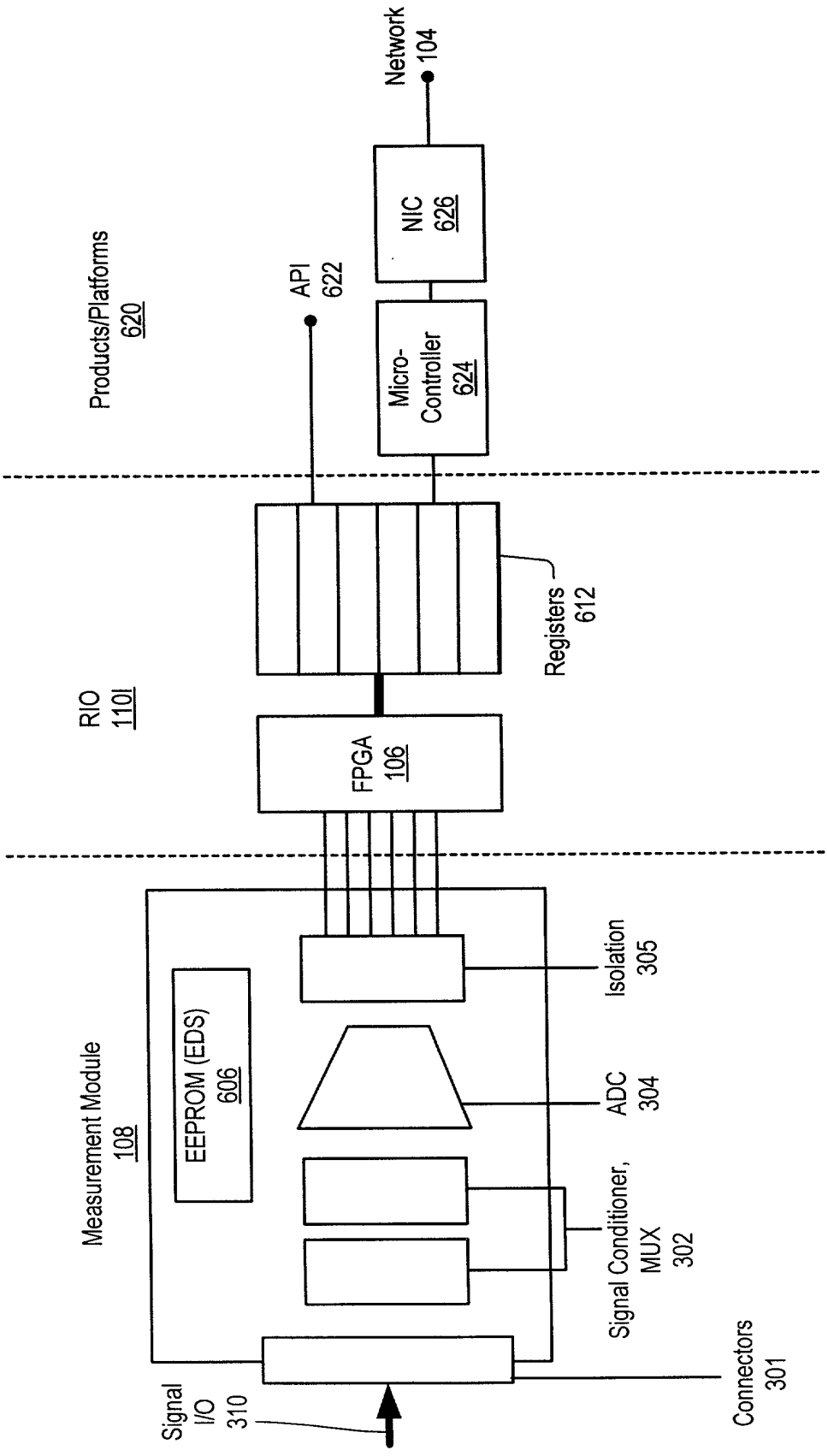

Network
104
NIC
626
Micro-
Controller
624
API
622
Products/Platforms
620
RIO
110I
FPGA
106
Registers
612
Measurement Module
108
EEPROM (EDS)
606
Isolation
305
ADC
304
Signal Conditioner,
MUX
302
Signal
I/O
310
Connectors
301

Figure 6

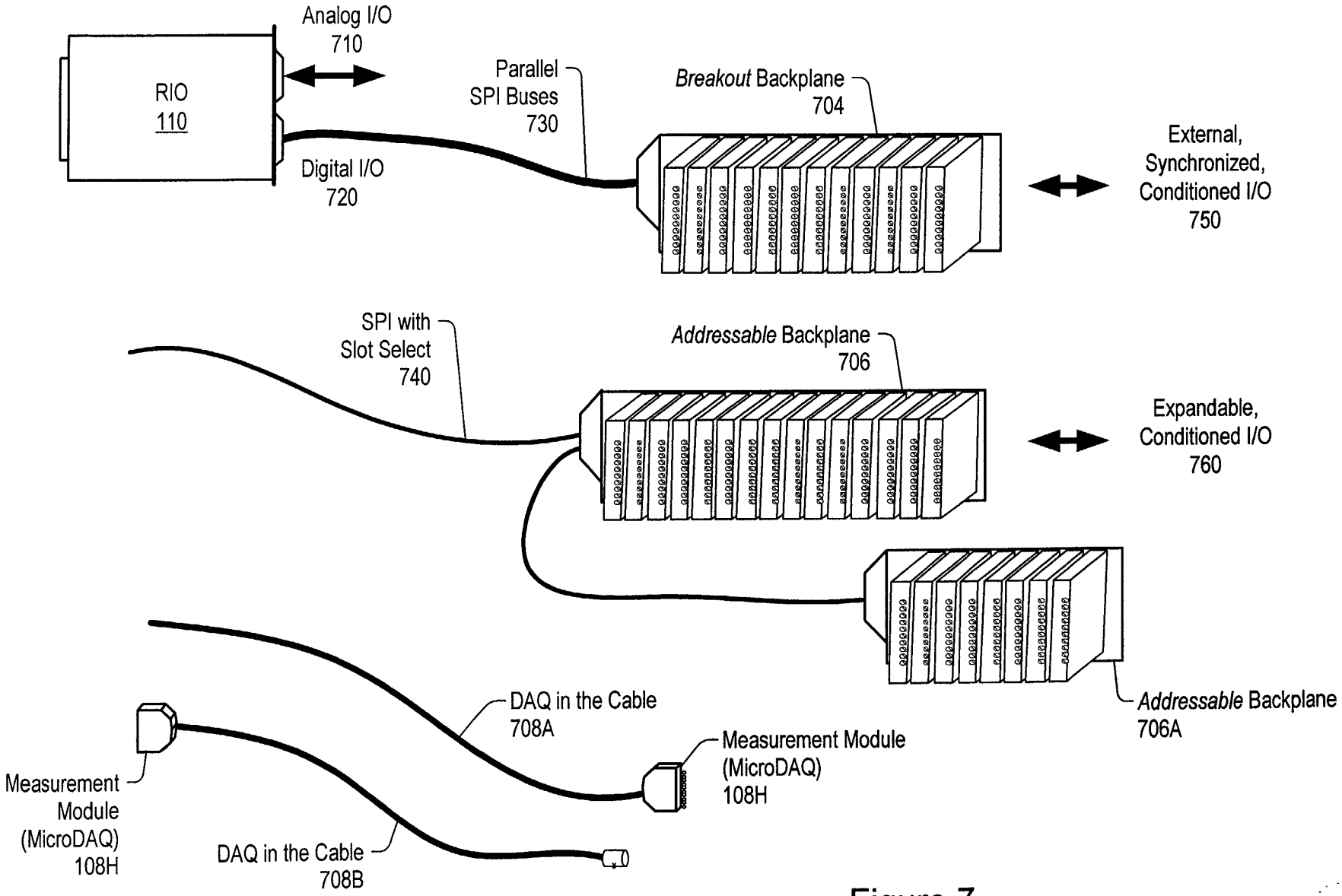

Analog I/O
710
RIO
110
Digital I/O
720
Parallel
SPI Buses
730
Breakout Backplane
704
External,
Synchronized,
Conditioned I/O
750
SPI with
Slot Select
740
Addressable Backplane
706
Expandable,
Conditioned I/O
760
Addressable Backplane
706A
DAQ in the Cable
708A
Measurement Module
(MicroDAQ)
108H
Measurement
Module
(MicroDAQ)
108H
DAQ in the Cable
708B

Figure 7

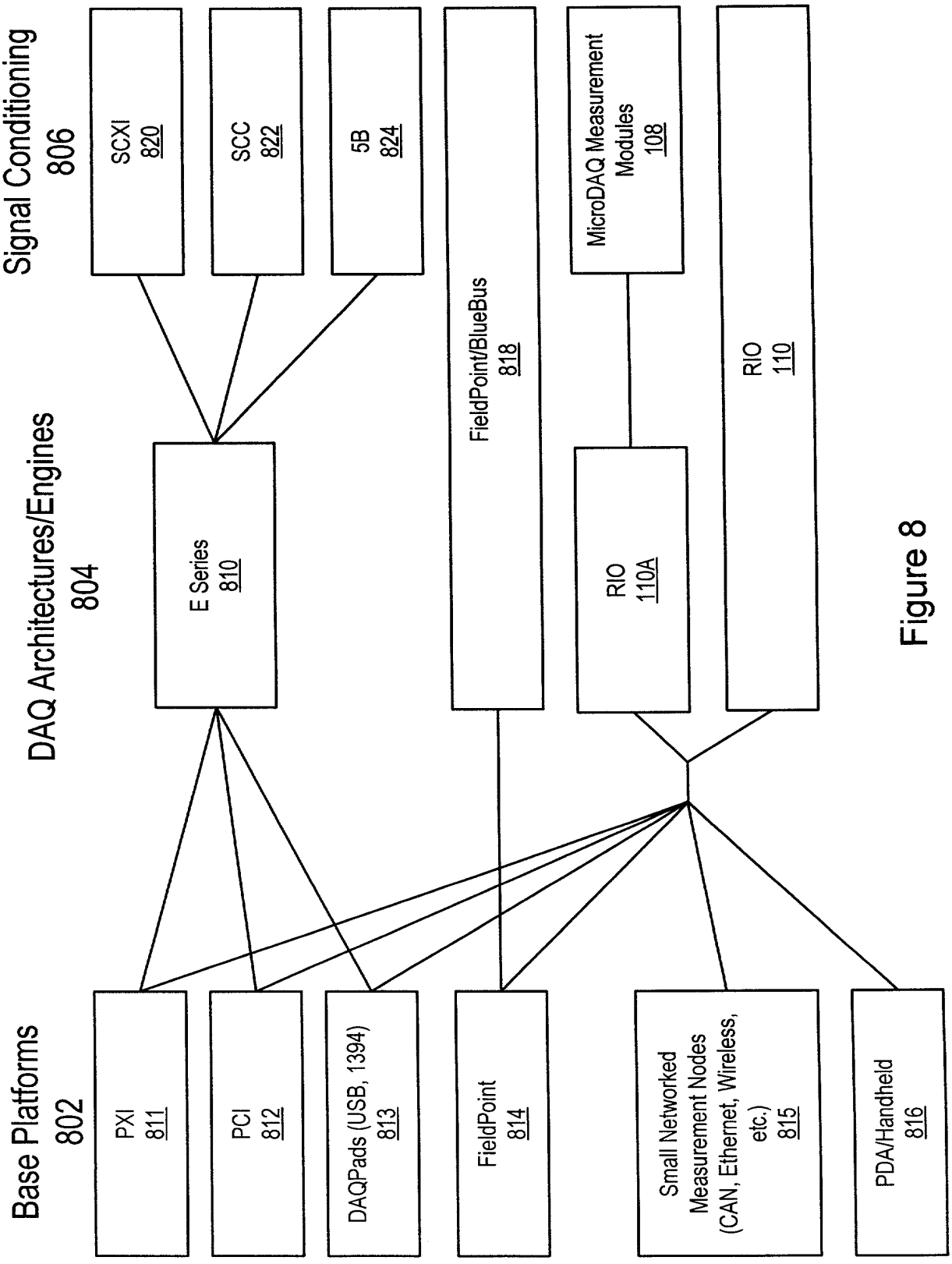

Base Platforms
802
PXI
811
PCI
812
DAQPads (USB, 1394)
813
FieldPoint
814
Small Networked
Measurement Nodes
(CAN, Ethernet, Wireless,
etc.)
815
PDA/Handheld
816
DAQ Architectures/Engines
804
E Series
810
FieldPoint/BlueBus
818
RIO
110A
RIO
110
Signal Conditioning
806
SCXI
820
SCC
822
5B
824
MicroDAQ Measurement
Modules
108

Figure 8

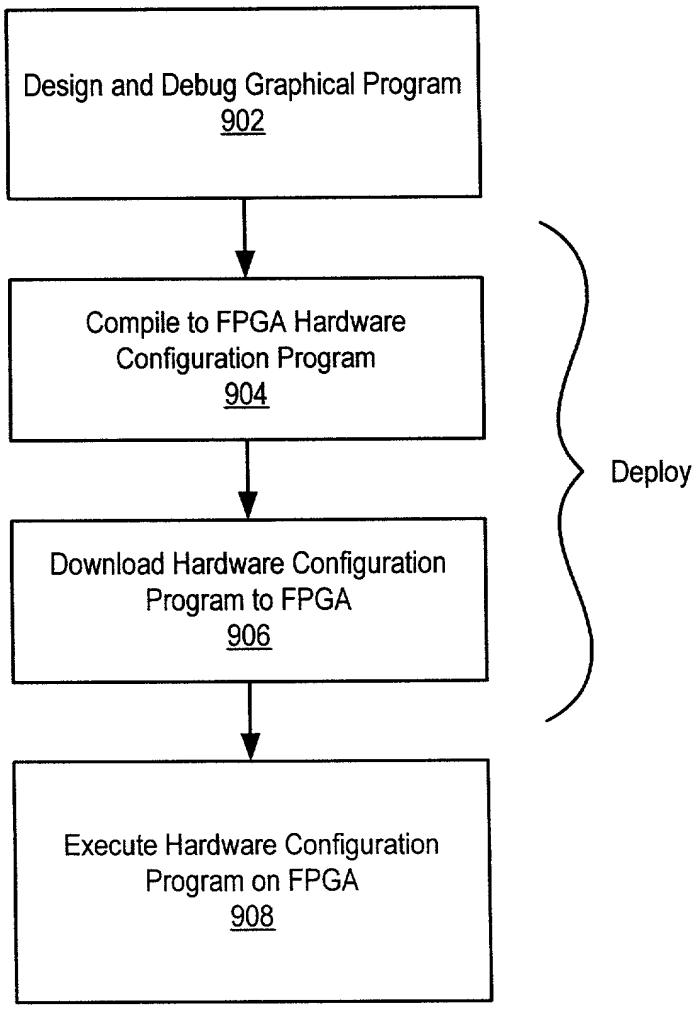

Design and Debug Graphical Program
902
Compile to FPGA Hardware Configuration Program
904
Download Hardware Configuration Program to FPGA
906
Execute Hardware Configuration Program on FPGA
908
Deploy

Figure 9

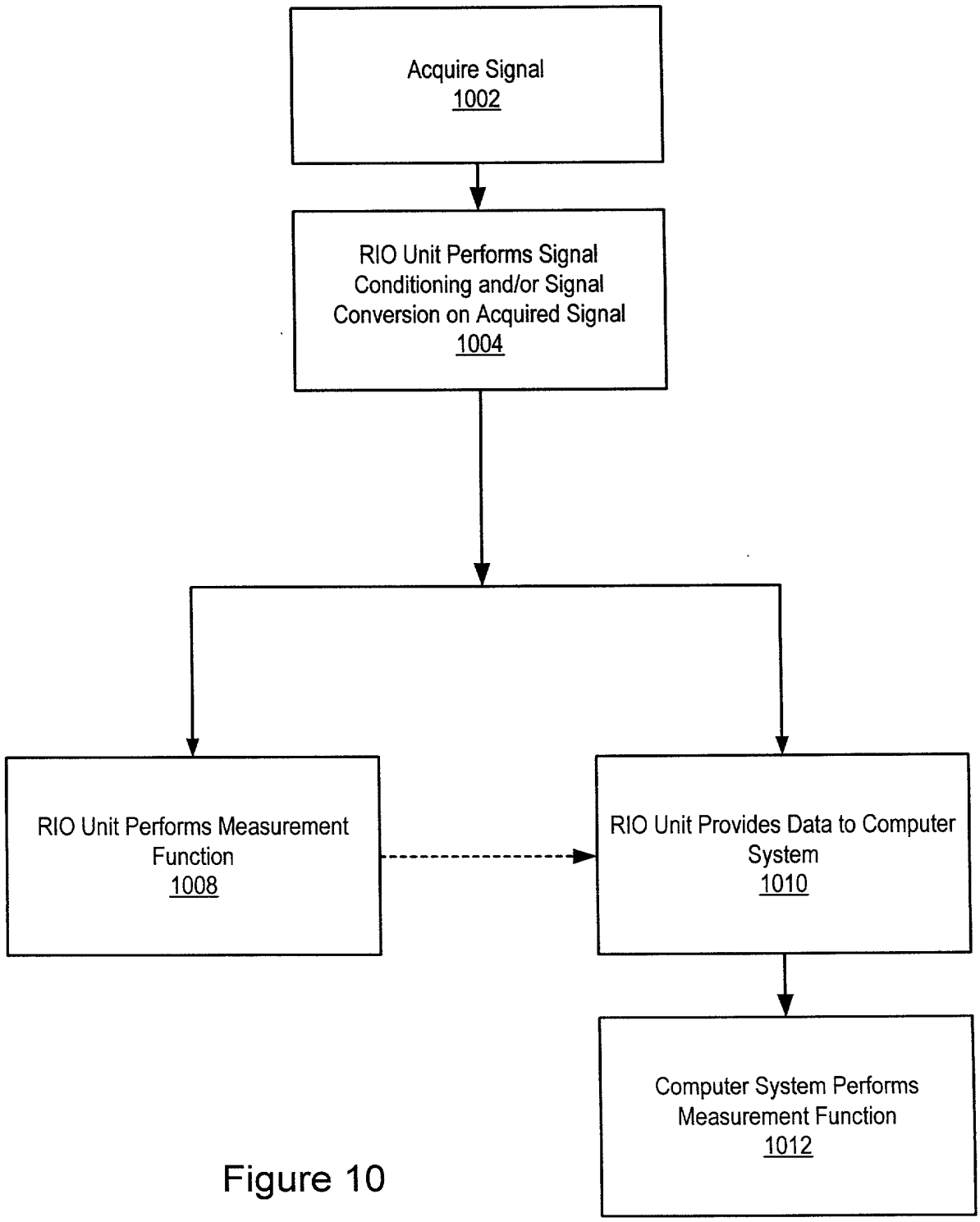

Acquire Signal
1002
RIO Unit Performs Signal Conditioning and/or Signal Conversion on Acquired Signal
1004
RIO Unit Performs Measurement Function
1008
RIO Unit Provides Data to Computer System
1010
Computer System Performs Measurement Function
1012

Figure 10

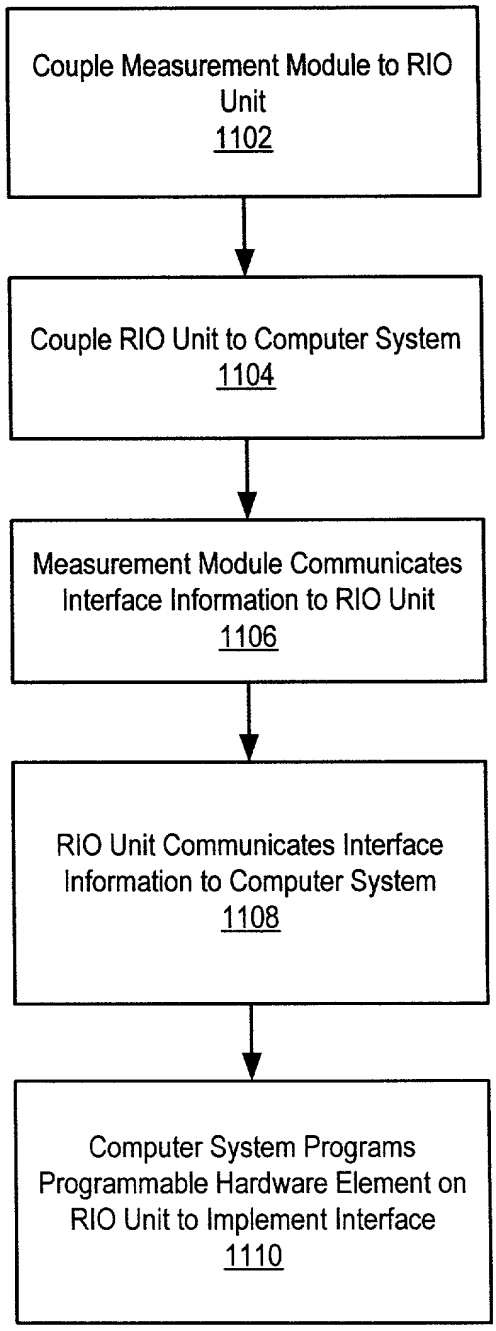

Couple Measurement Module to RIO Unit
1102
Couple RIO Unit to Computer System
1104
Measurement Module Communicates Interface Information to RIO Unit
1106
RIO Unit Communicates Interface Information to Computer System
1108
Computer System Programs Programmable Hardware Element on RIO Unit to Implement Interface
1110

Figure 11A

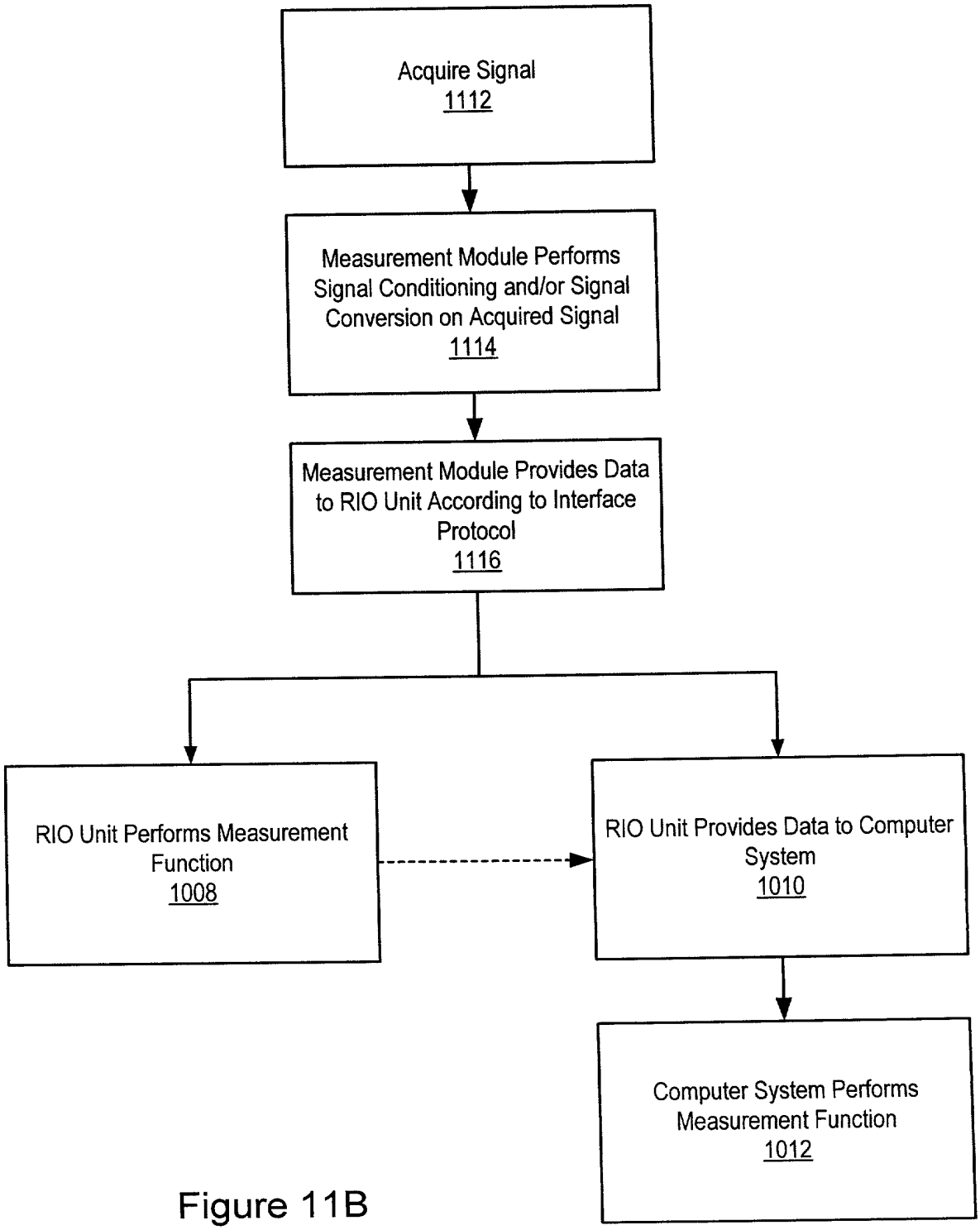

Acquire Signal
1112
Measurement Module Performs Signal Conditioning and/or Signal Conversion on Acquired Signal
1114
Measurement Module Provides Data to RIO Unit According to Interface Protocol
1116
RIO Unit Performs Measurement Function
1008
RIO Unit Provides Data to Computer System
1010
Computer System Performs Measurement Function
1012

Figure 11B

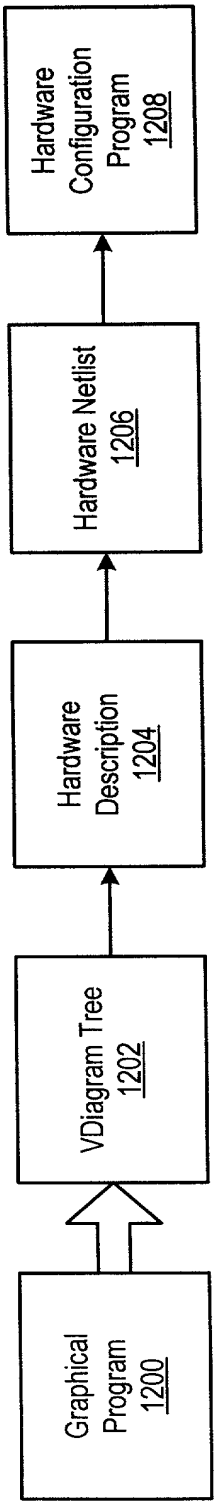

Graphical Program 1200
VDiagram Tree 1202
Hardware Description 1204
Hardware Netlist 1206
Hardware Configuration Program 1208

Figure 12

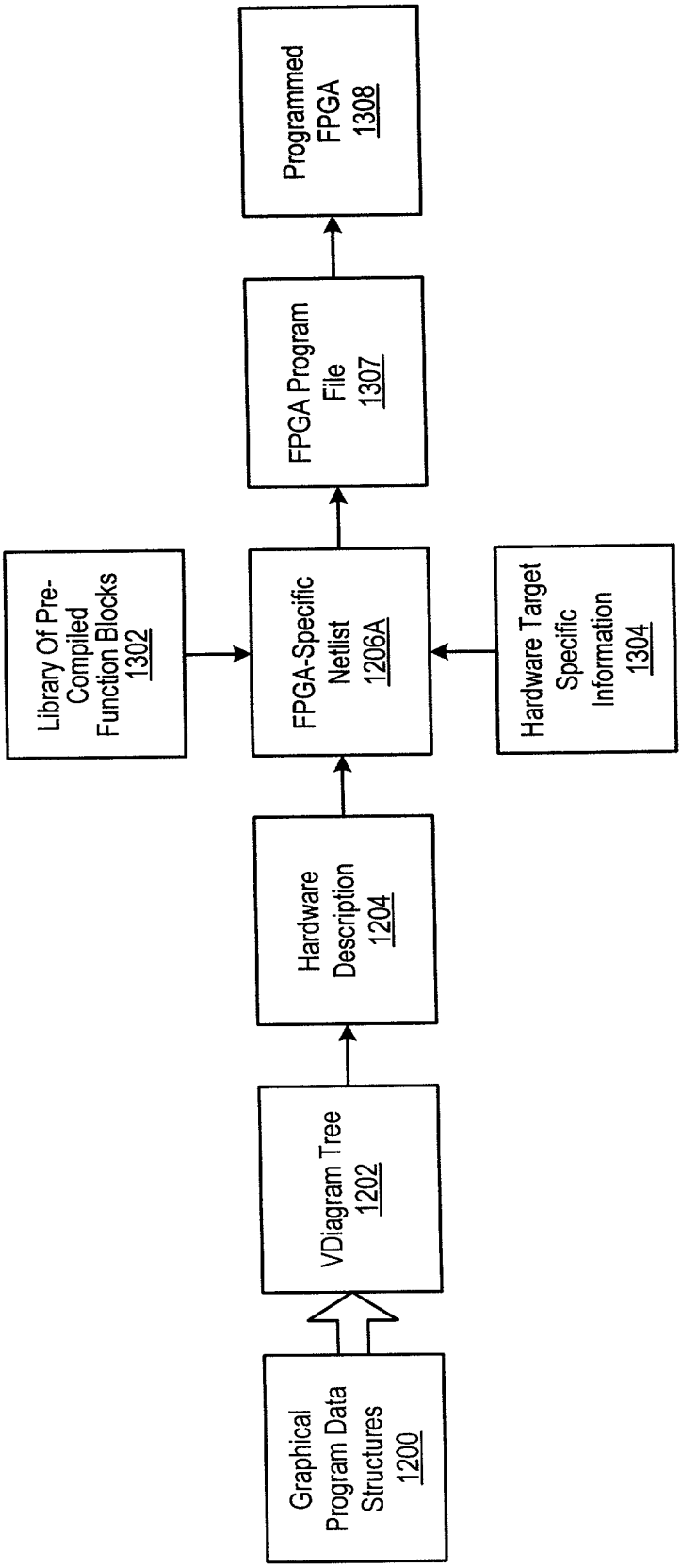

Graphical Program Data Structures 1200
VDiagram Tree 1202
Hardware Description 1204
Library Of Pre-Compiled Function Blocks 1302
FPGA-Specific Netlist 1206A
Hardware Target Specific Information 1304
FPGA Program File 1307
Programmed FPGA 1308

Figure 13

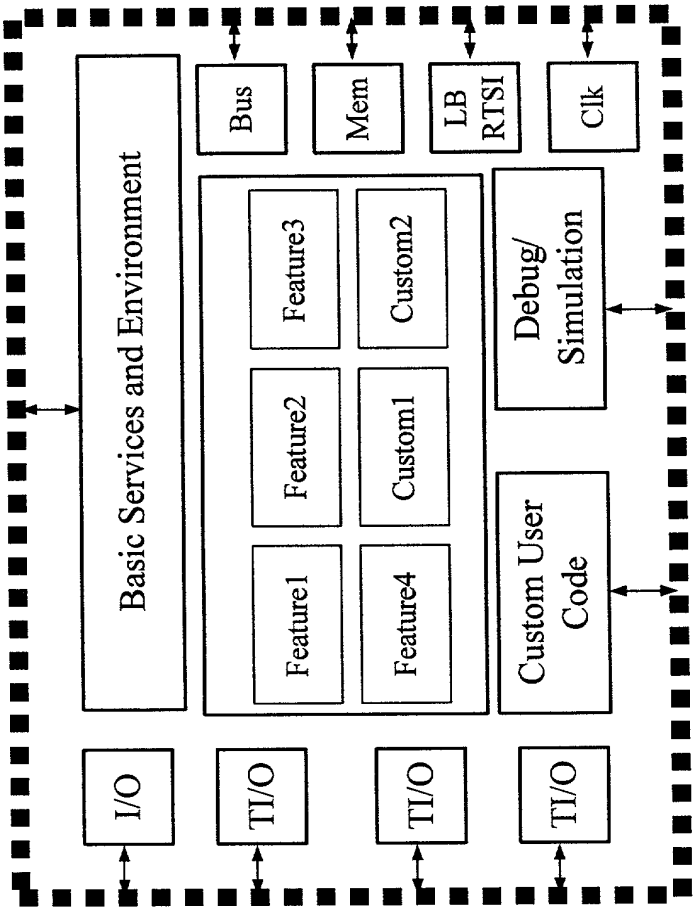

Basic Services and Environment
Bus
Mem
LB RTSI
Clk
Feature1
Feature2
Feature3
Feature4
Custom1
Custom2
Debug/ Simulation
Custom User Code
I/O
TI/O
TI/O
TI/O

Figure 14

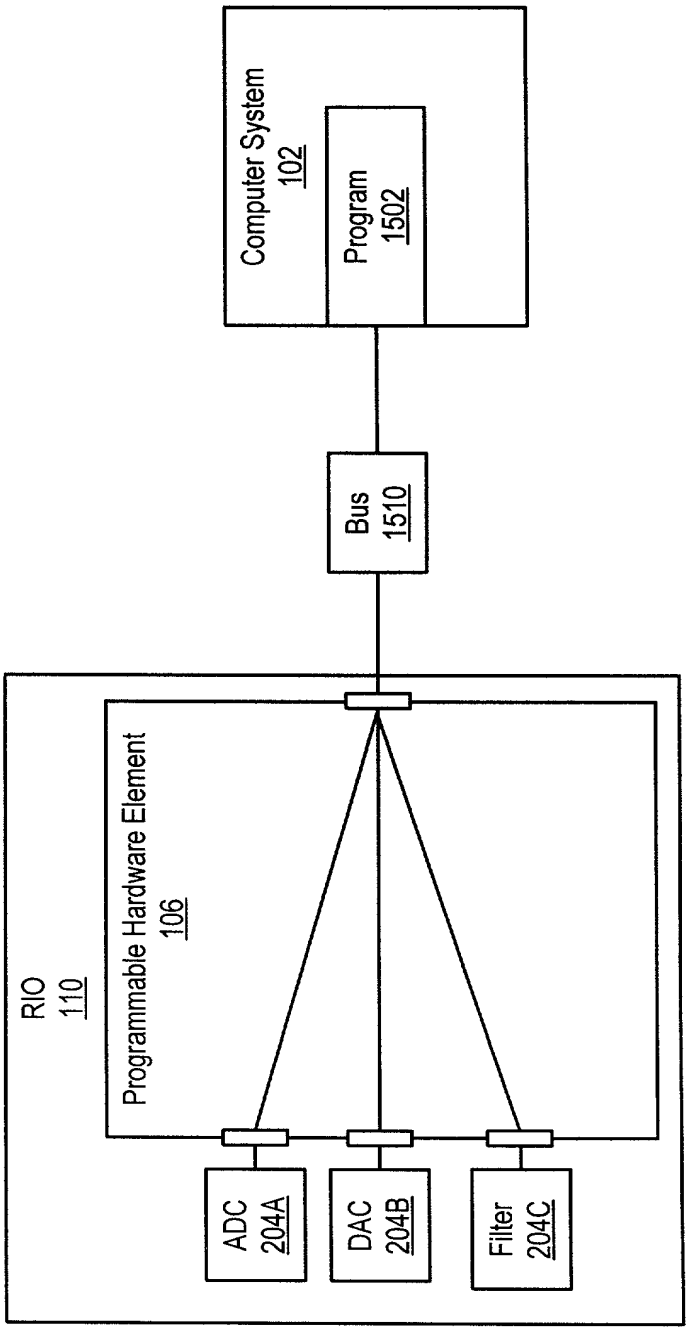

Computer System
102
Program
1502
Bus
1510
RIO
110
Programmable Hardware Element
106
ADC
204A
DAC
204B
Filter
204C

Figure 15

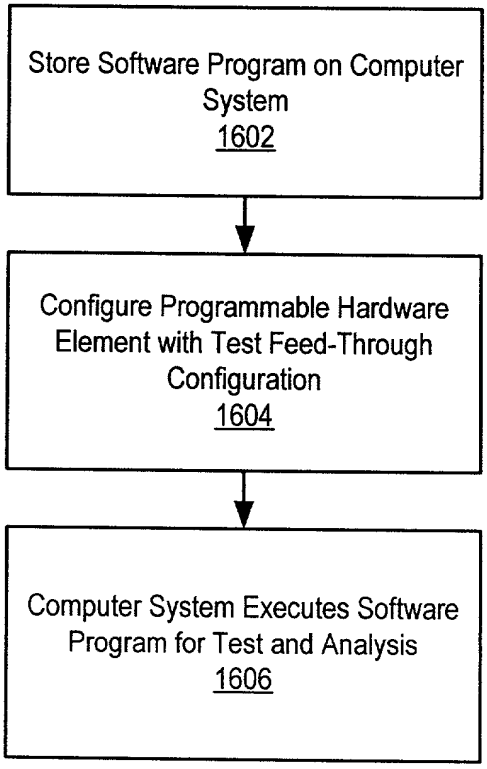

Store Software Program on Computer System
1602
Configure Programmable Hardware Element with Test Feed-Through Configuration
1604
Computer System Executes Software Program for Test and Analysis
1606

Figure 16

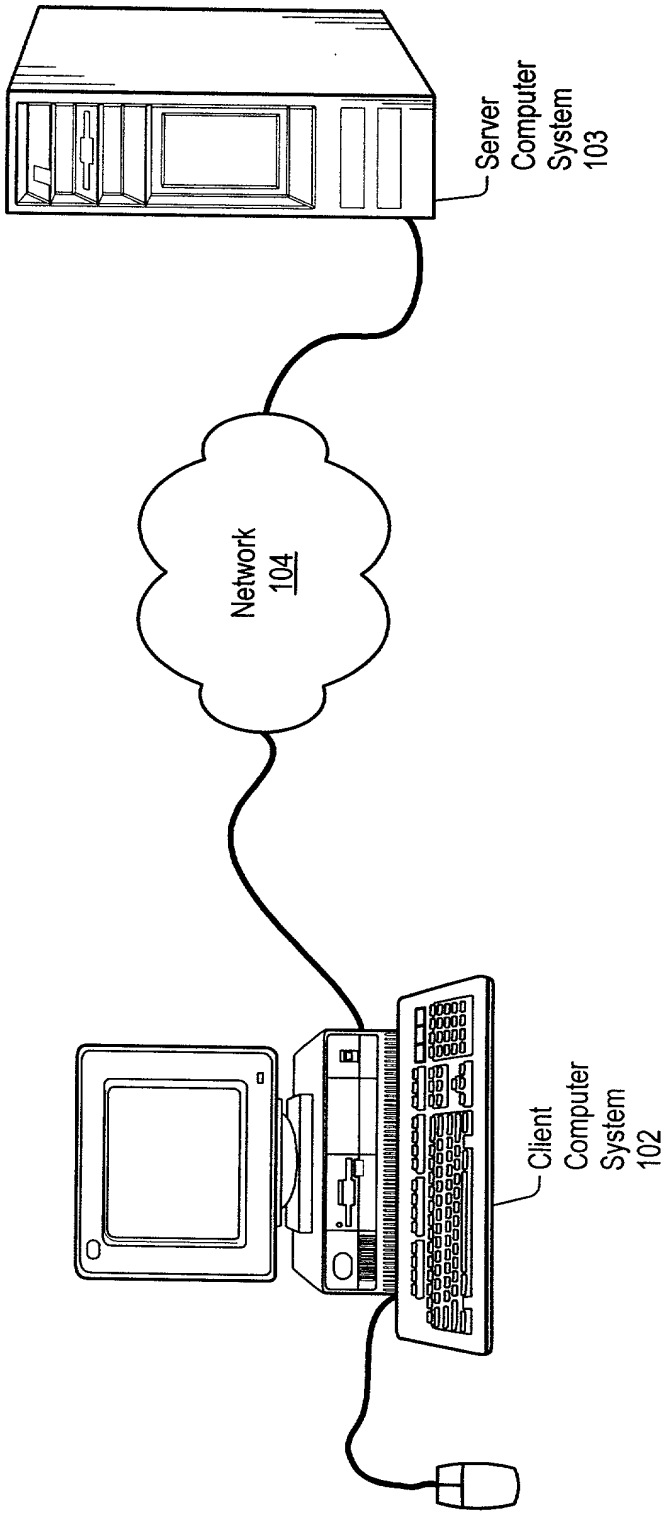

Server
Computer
System
103
Network
104
Client
Computer
System
102

Figure 17

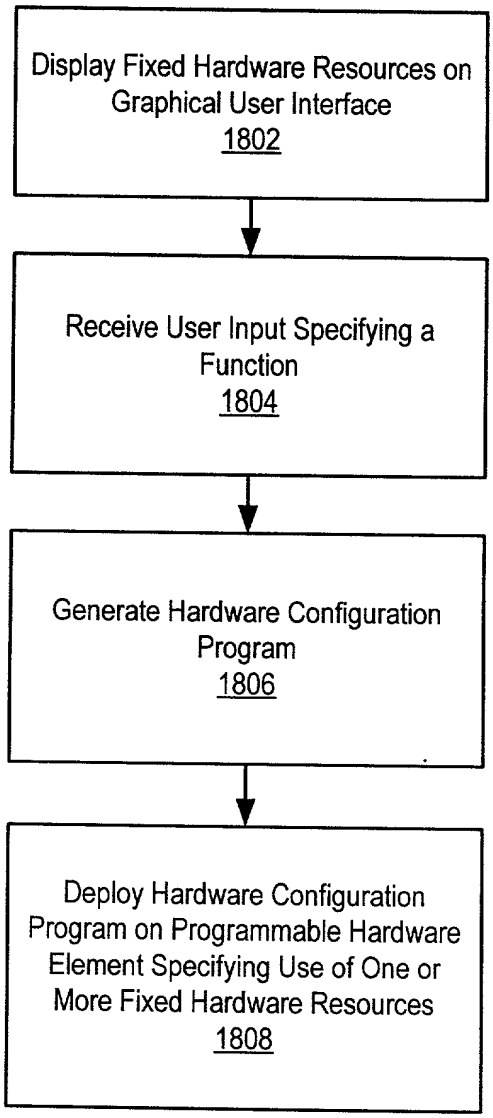

Display Fixed Hardware Resources on Graphical User Interface
1802
Receive User Input Specifying a Function
1804
Generate Hardware Configuration Program
1806
Deploy Hardware Configuration Program on Programmable Hardware Element Specifying Use of One or More Fixed Hardware Resources
1808

Figure 18

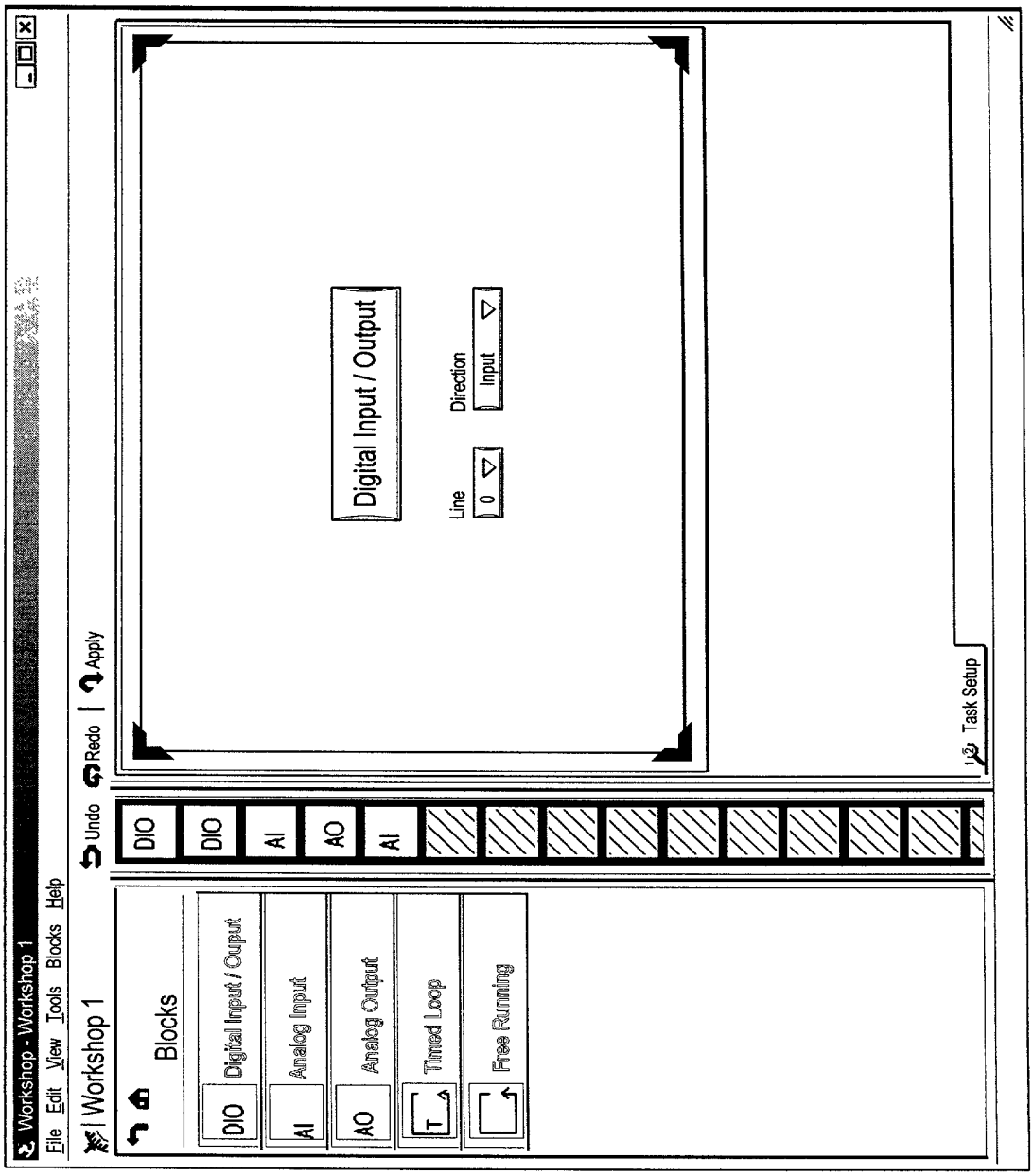

Workshop - Workshop 1
File Edit View Tools Blocks Help
Workshop 1
Undo
Redo
Apply
Blocks
DIO
Digital Input / Output
AI
Analog Input
AO
Analog Output
T
Timed Loop
Free Running
DIO
DIO
AI
AO
AI
Digital Input / Output
Line
0
Direction
Input
Task Setup

Figure 19A

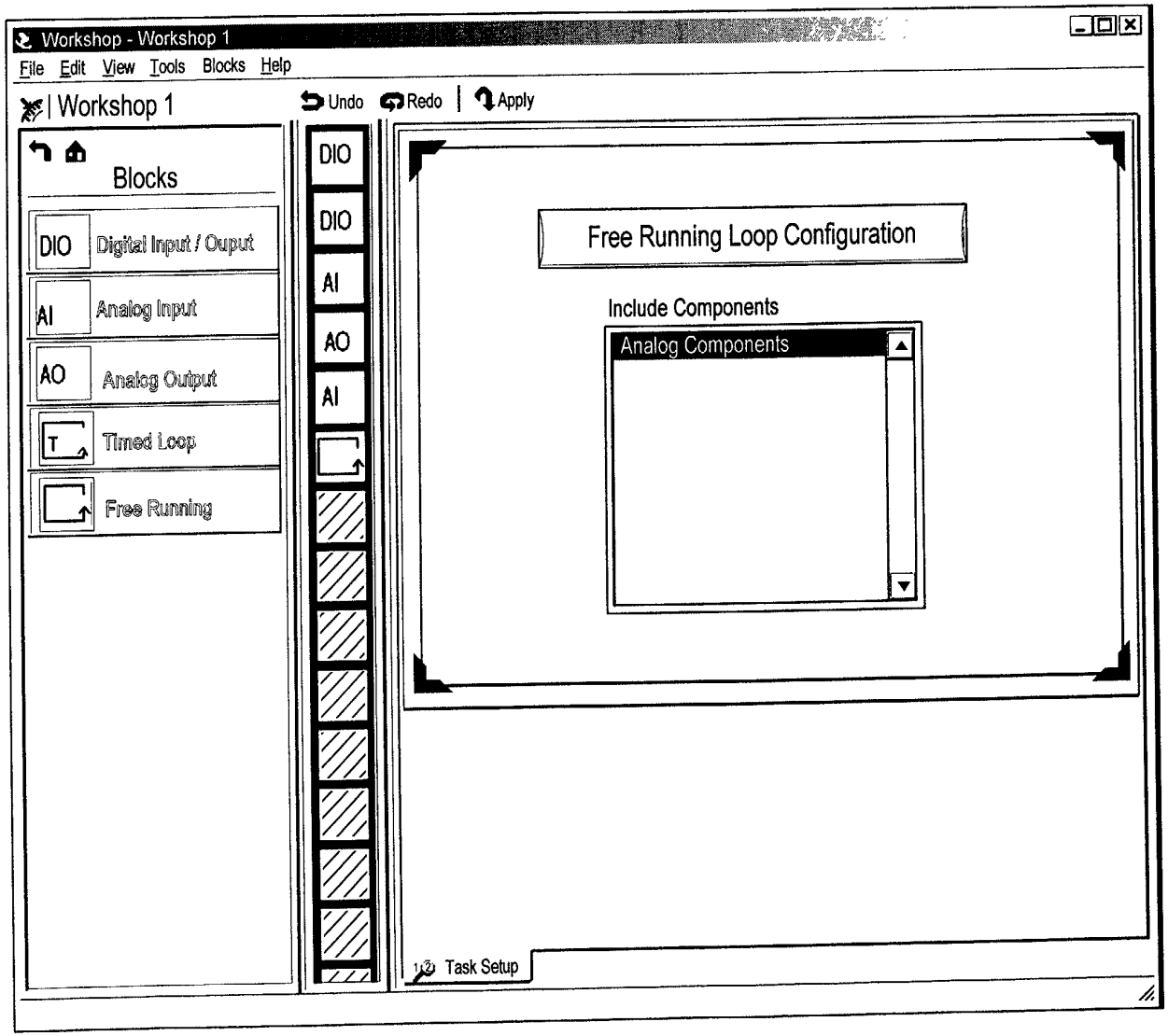
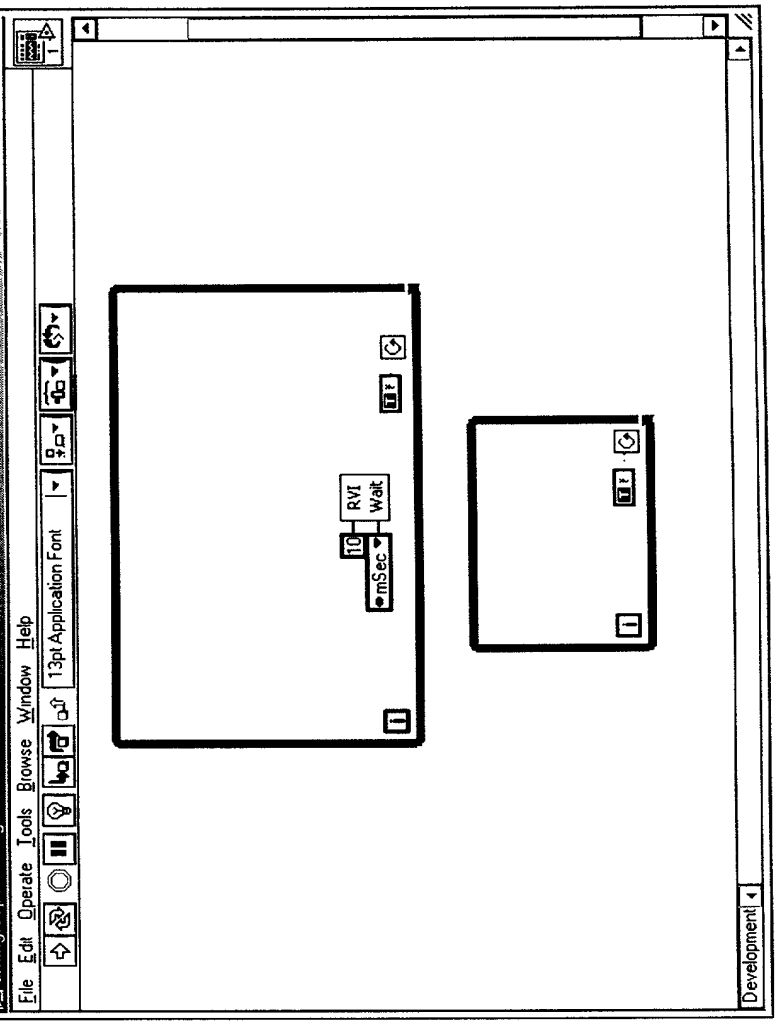
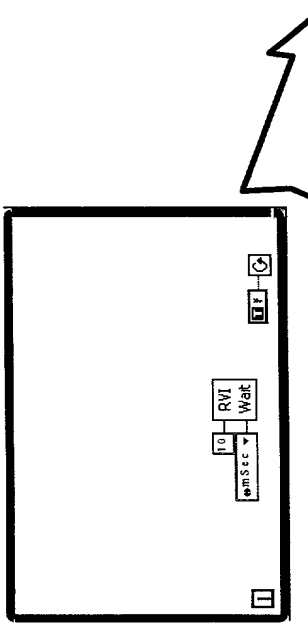

Workshop - Workshop 1
File Edit View Tools Blocks Help
Workshop 1
Undo
Redo
Apply
Blocks
DIO
Digital Input / Ouput
AI
Analog Input
AO
Analog Output
T
Timed Loop
Free Running
DIO
DIO
AI
AO
AI
Free Running Loop Configuration
Include Components
Analog Components
Task Setup

Figure 19B


timingloops.vi Diagram
File Edit Operate Tools Browse Window Help
13pt Application Font
RVI Wait
mSec
Development
Diagram


RVI Wait
mSec
Timed Loop

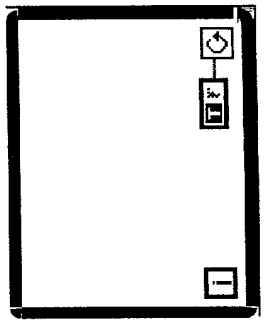

Free Running Loop

Figure 19C

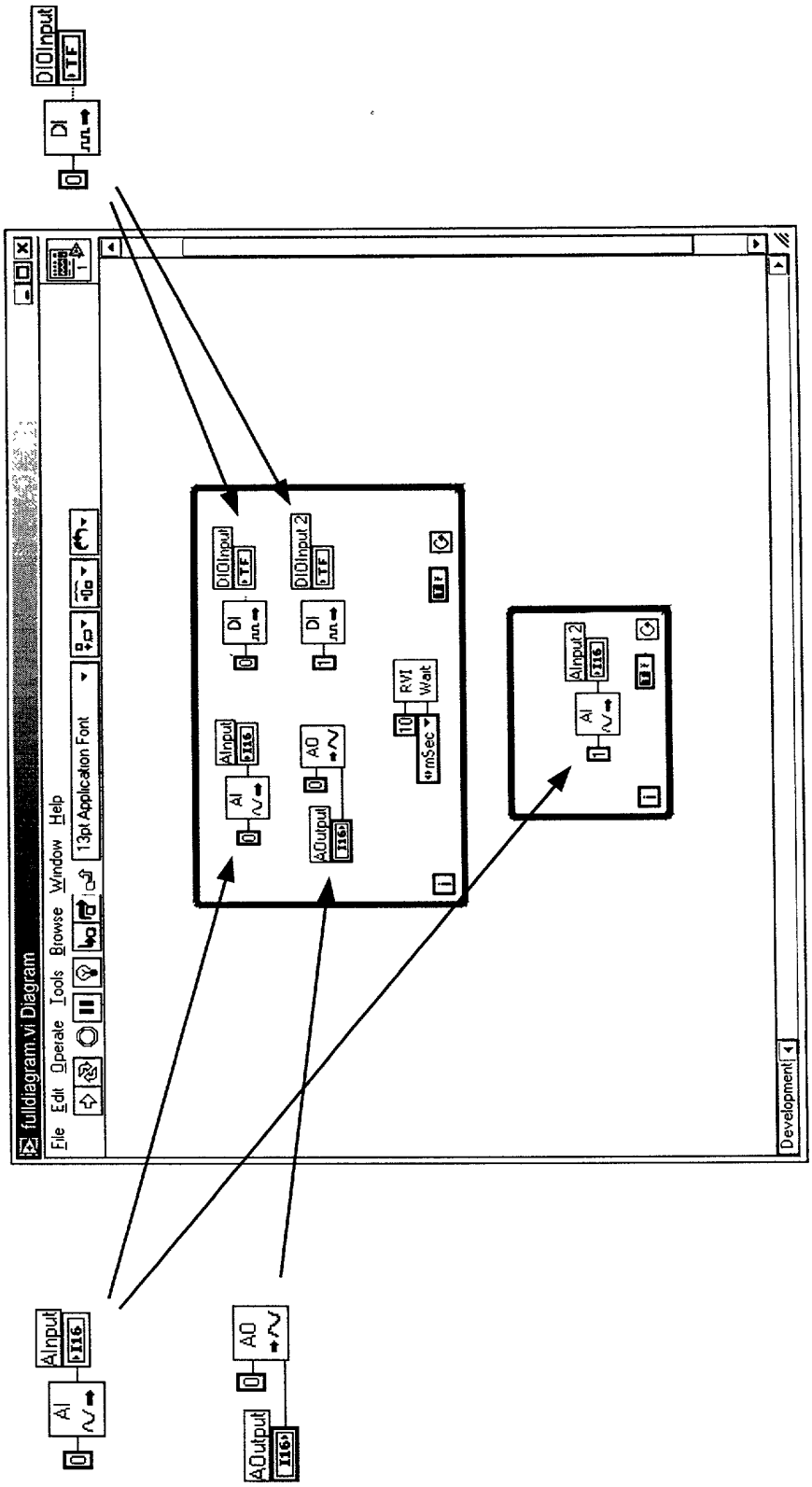

fulldiagram.vi Diagram
File Edit Operate Tools Browse Window Help
13pt Application Font
Development
DIOInput
TF
DI
AInput
116
AI
AO
AOutput
RVI Wait
10
mSec
DIOInput 2
AInput 2

Figure 19D

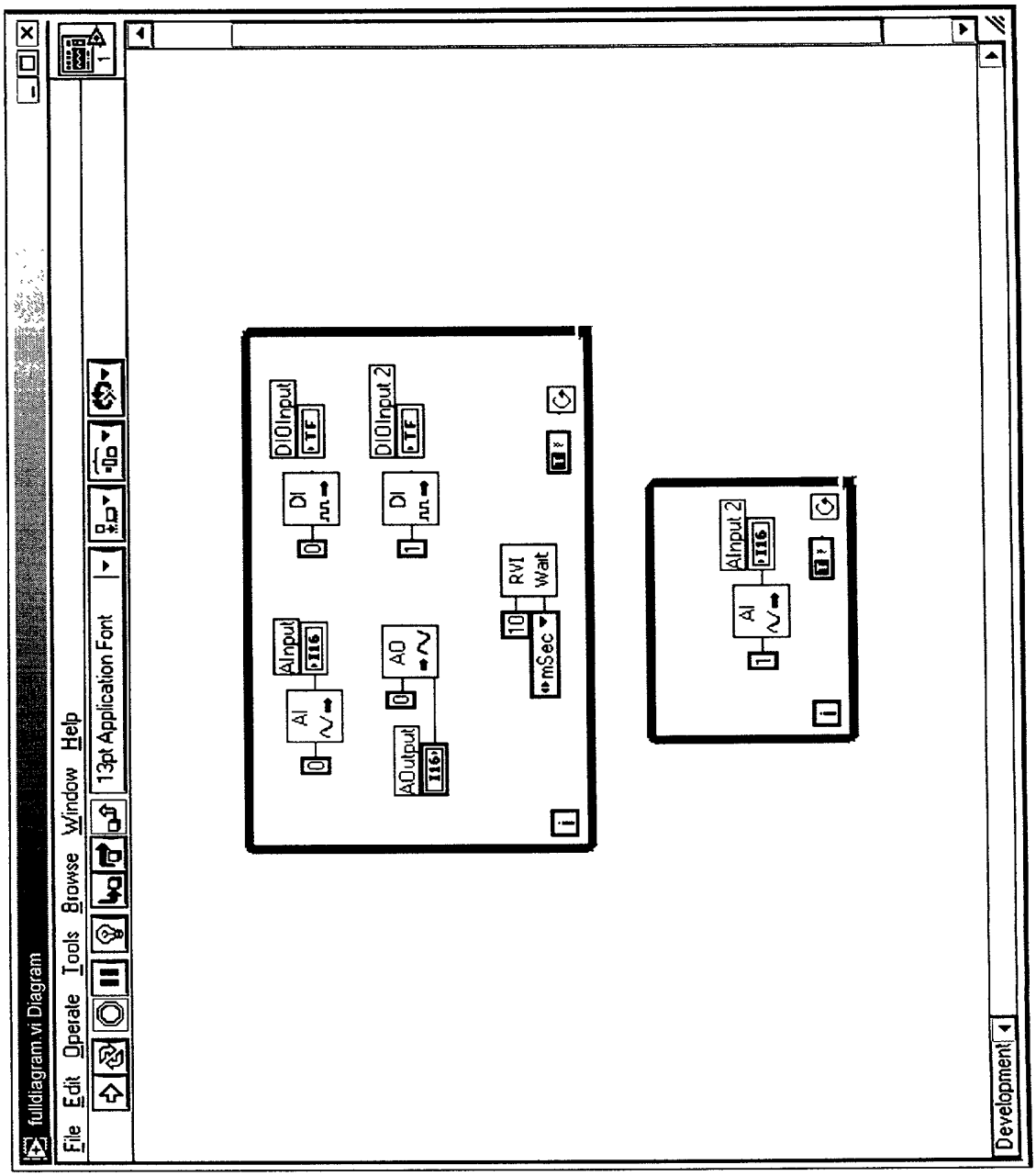

fulldiagram.vi Diagram
File Edit Operate Tools Browse Window Help
13pt Application Font
AI
AInput
AO
AOutput
DI
DIOInput
DI
DIOInput 2
10
RVI Wait
mSec
AI
AInput 2
Development

Figure 19E

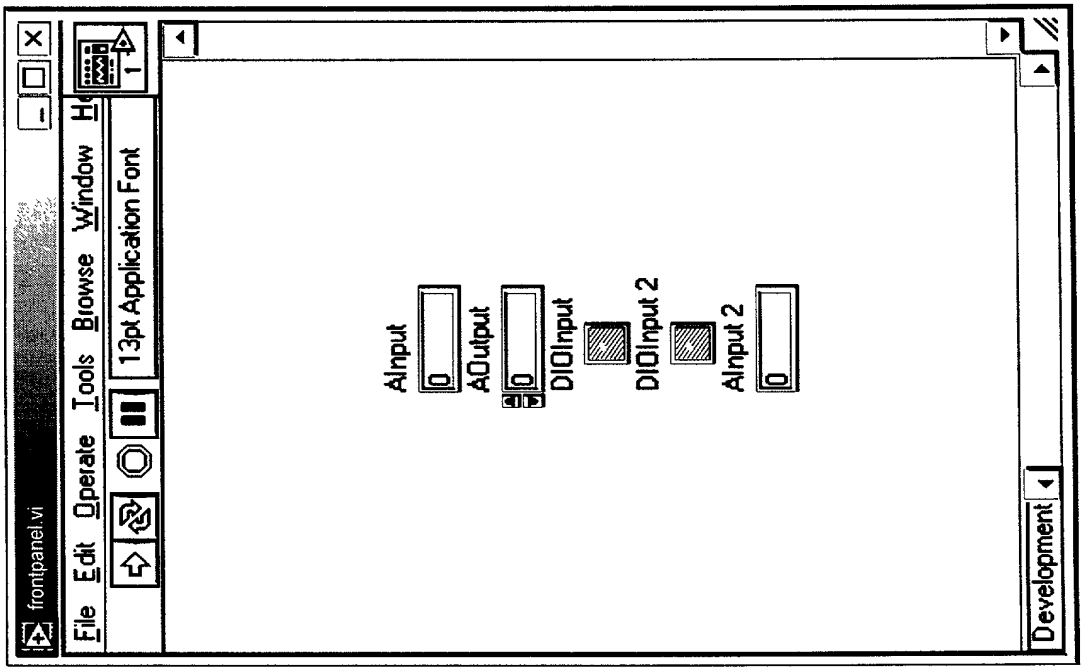

frontpanel.vi
File Edit Operate Tools Browse Window Help
13pt Application Font
AInput
AOutput
DIOInput
DIOInput 2
AInput 2
Development

Figure 19F

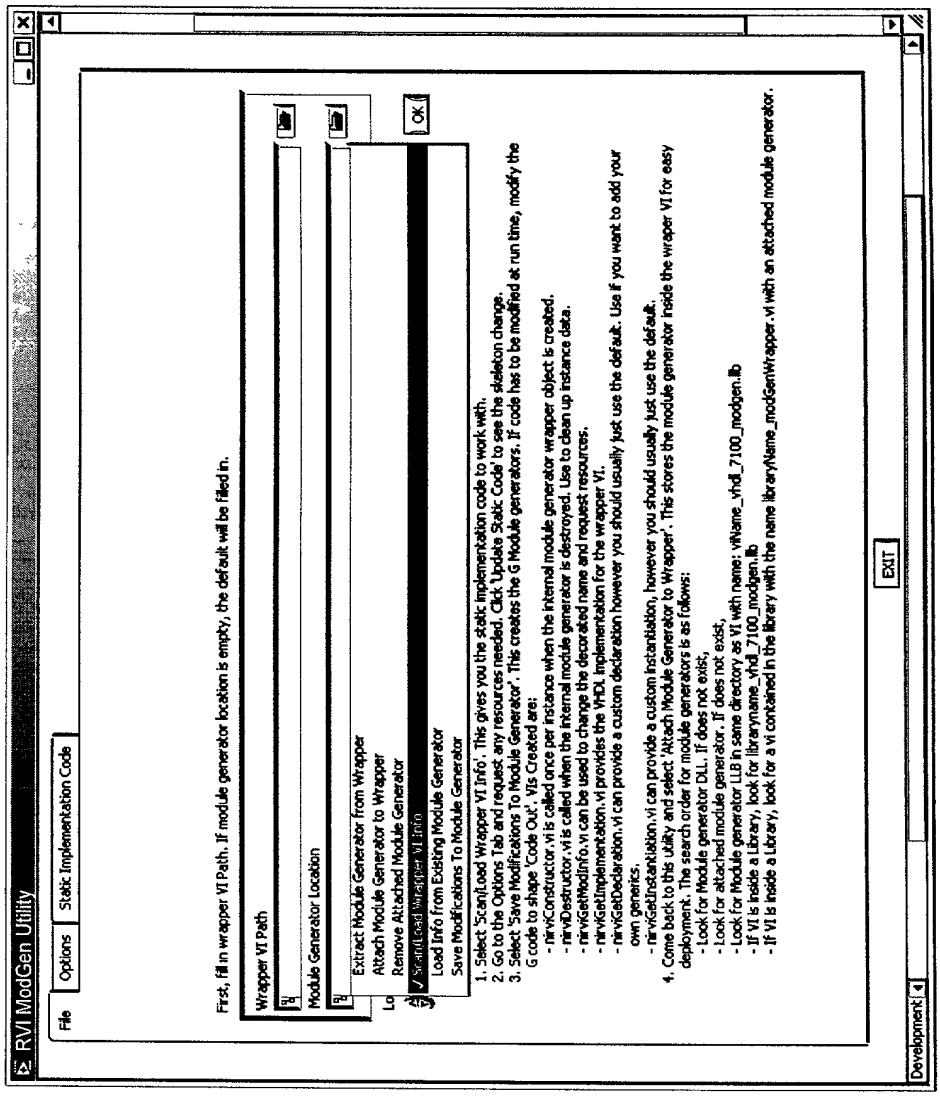

RVI ModGen Utility
File
Options
Static Implementation Code
First, fill in wrapper VI Path. If module generator location is empty, the default will be filled in.
Wrapper VI Path
Module Generator Location
Extract Module Generator from Wrapper
Attach Module Generator to Wrapper
Remove Attached Module Generator
Scan/Load Wrapper VI Info
Load Info from Existing Module Generator
Save Modifications To Module Generator
OK
1. Select 'Scan/Load Wrapper VI Info'. This gives you the static implementation code to work with.
2. Go to the Options Tab and request any resources needed. Click 'Update Static Code' to see the skeleton change.
3. Select 'Save Modifications To Module Generator'. This creates the G Module generators. If code has to be modified at run time, modify the G code to shape 'Code Out'. VIs Created are:
- nirviConstructor.vi is called once per instance when the internal module generator wrapper object is created.
- nirviDestructor.vi is called when the internal module generator is destroyed. Use to clean up instance data.
- nirviGetModInfo.vi can be used to change the decorated name and request resources.
- nirviGetImplementation.vi provides the VHDL implementation for the wrapper VI.
- nirviGetDeclaration.vi can provide a custom declaration however you should usually just use the default. Use if you want to add your own generics.
- nirviGetInstantiation.vi can provide a custom instantiation, however you should usually just use the default.
4. Come back to this utility and select 'Attach Module Generator to Wrapper'. This stores the module generator inside the wrapper VI for easy deployment. The search order for module generators is as follows:
- Look for Module generator DLL. If does not exist,
- Look for attached module generator. If does not exist,
- Look for Module generator LLB in same directory as VI with name: viName_vhdl_7100_modgen.llb
- If VI is inside a Library, look for libraryname_vhdl_7100_modgen.llb
- If VI is inside a Library, look for a vi contained in the library with the name libraryName_modGenWrapper.vi with an attached module generator.
EXIT
Development

Figure 20

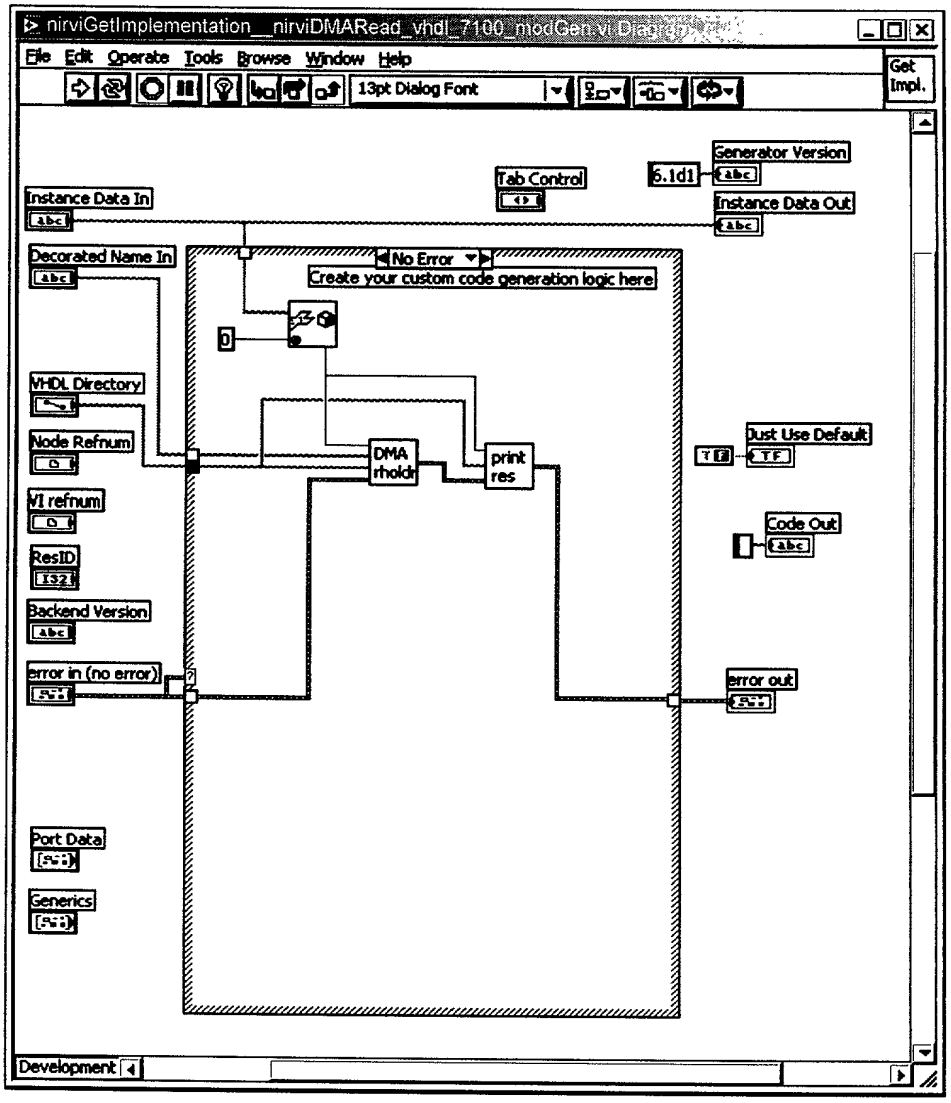

nirviGetImplementation__nirviDMARead_vhdl_7100_modGen.vi Diagram
File Edit Operate Tools Browse Window Help
13pt Dialog Font
Get Impl.
Generator Version
6.1d1
Tab Control
Instance Data In
Instance Data Out
Decorated Name In
No Error
Create your custom code generation logic here
0
VHDL Directory
Node Refnum
DMA rholdr
print res
Dust Use Default
TF
VI refnum
Code Out
ResID
I32
Backend Version
error in (no error)
error out
Port Data
Generics
Development

Figure 21

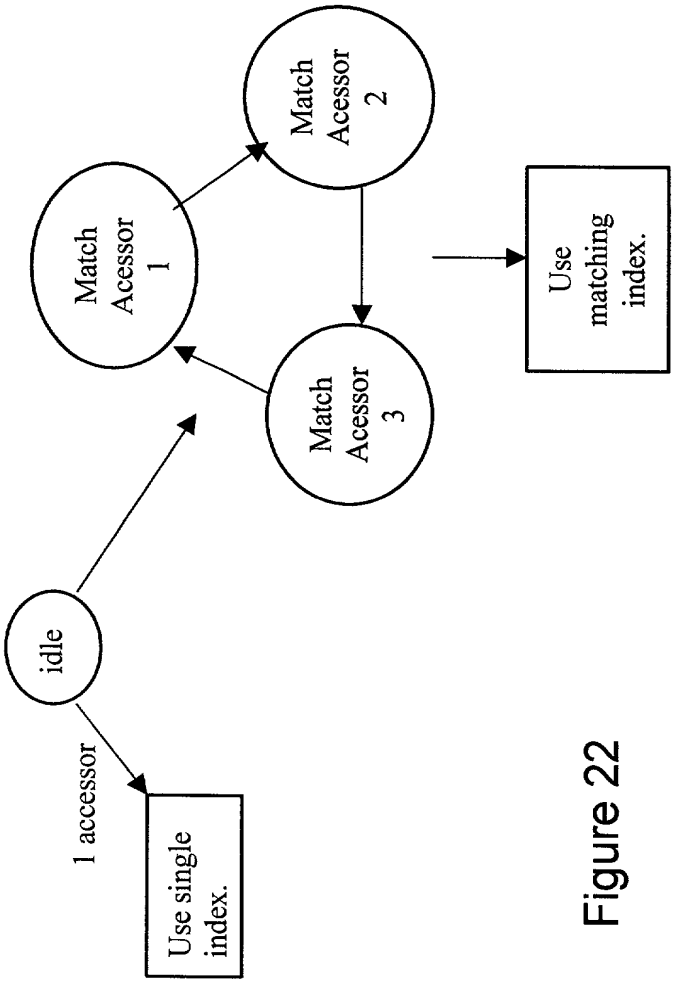

idle
1 accessor
Use single index.
Match Acessor 1
Match Acessor 2
Match Acessor 3
Use matching index.

Figure 22

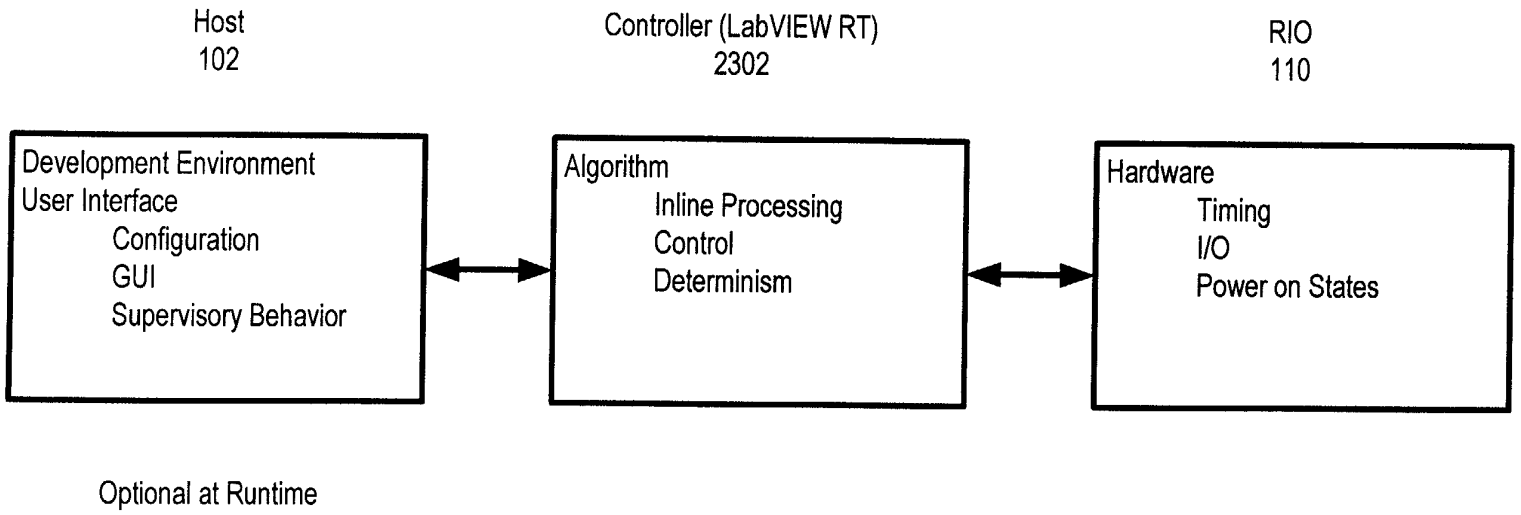

Host
102
Controller (LabVIEW RT)
2302
RIO
110
Development Environment
User Interface
Configuration
GUI
Supervisory Behavior
Algorithm
Inline Processing
Control
Determinism
Hardware
Timing
I/O
Power on States
Optional at Runtime

Figure 23A

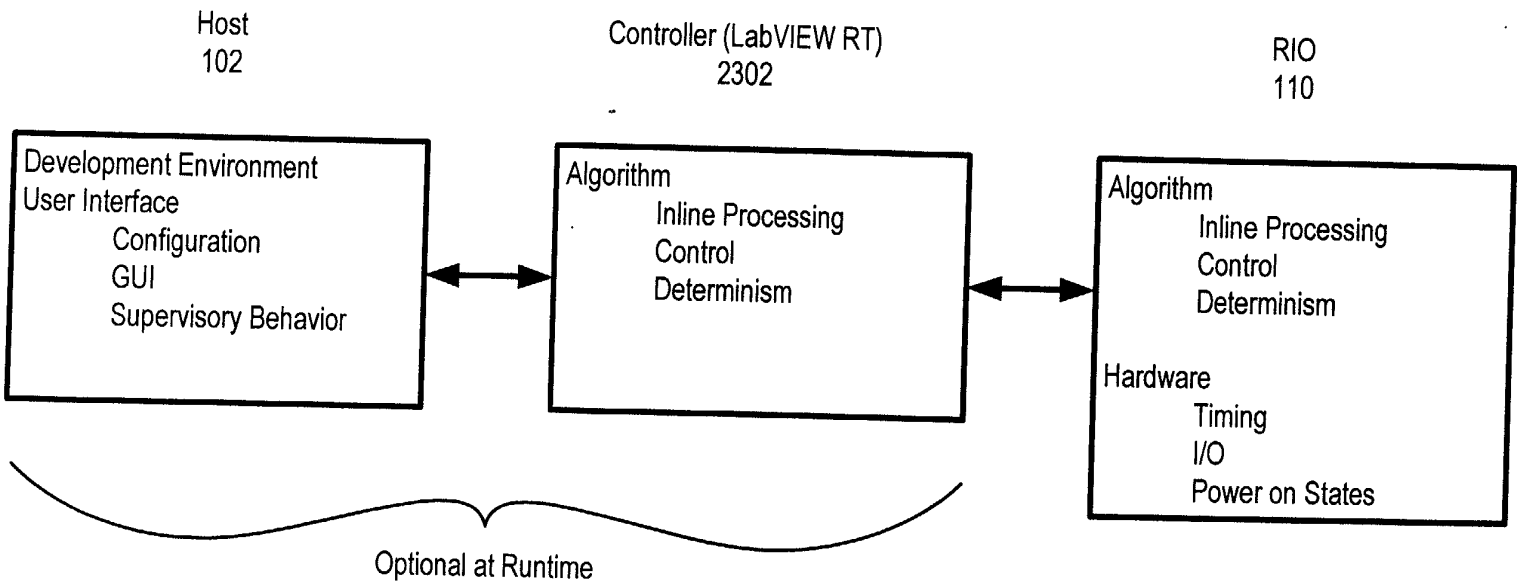

Host
102
Controller (LabVIEW RT)
2302
RIO
110
Development Environment
User Interface
Configuration
GUI
Supervisory Behavior
Algorithm
Inline Processing
Control
Determinism
Algorithm
Inline Processing
Control
Determinism
Hardware
Timing
I/O
Power on States
Optional at Runtime

Figure 23B

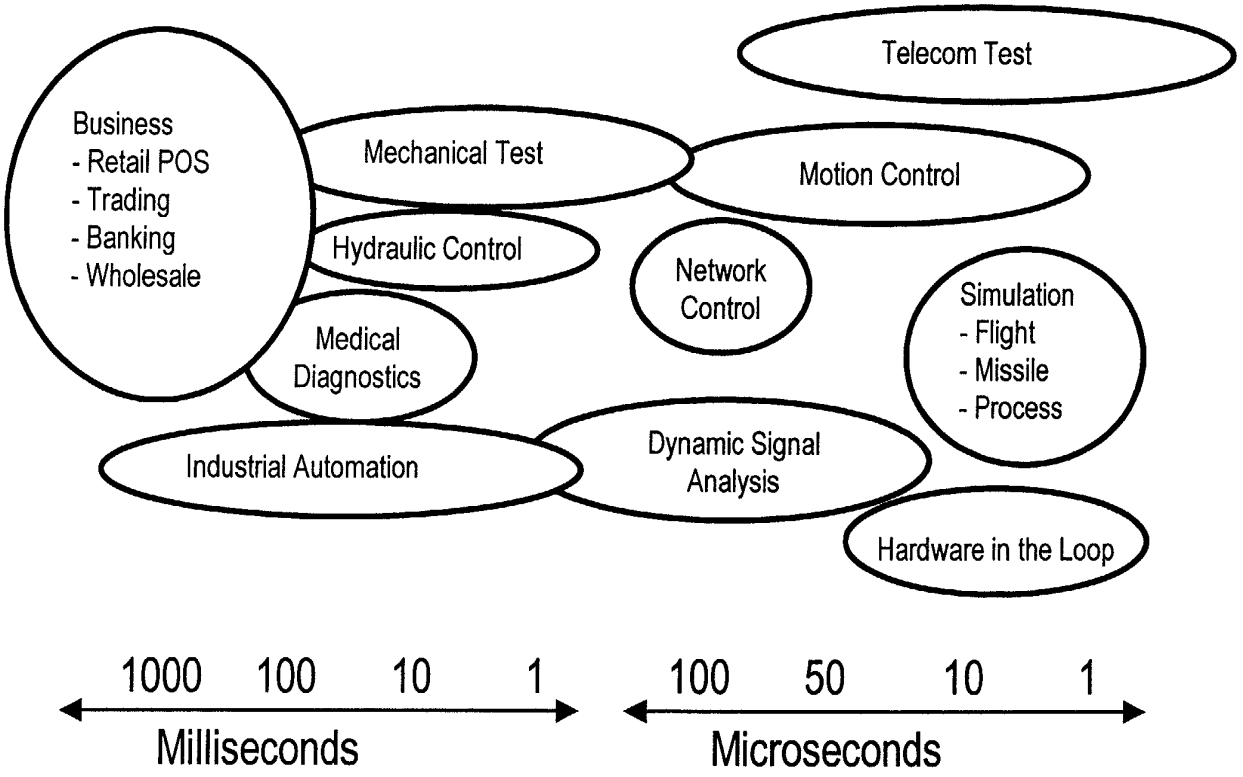

Telecom Test
Business
- Retail POS
- Trading
- Banking
- Wholesale
Mechanical Test
Motion Control
Hydraulic Control
Network Control
Simulation
- Flight
- Missile
- Process
Medical Diagnostics
Industrial Automation
Dynamic Signal Analysis
Hardware in the Loop
1000 100 10 1
Milliseconds
100 50 10 1
Microseconds

Figure 24